

---

# Entwicklung eines lidartypischen Objektlisten-Sensormodells

---

**Masterthesis Nr. 723/18**

Bearbeiter: Philip Aust | 1909867

Betreuer: Philipp Rosenberger, M.Sc.

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



---

Philip Aust  
Matrikelnummer: 1909867  
Studiengang: Mechatronik

Masterthesis Nr. 723/18  
Thema: Entwicklung eines lidartypischen Objektlisten-Sensormodells

Eingereicht: 29. Mai 2019

Technische Universität Darmstadt  
Fachgebiet Fahrzeugtechnik  
Prof. Dr. rer. nat. Hermann Winner  
Otto-Berndt-Straße 2  
64287 Darmstadt

---

---

Dieses Dokument wird bereitgestellt von TUpriints – Publikationsservice der TU Darmstadt.

<https://tuprints.ulb.tu-darmstadt.de/>

Bitte verweisen sie auf:

URN: urn:nbn:de:tuda-tuprints-89439

URI: <https://tuprints.ulb.tu-darmstadt.de/id/eprint/8943>

Lizenz: CC-BY-NC-ND 4.0 International

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

---

---

## **Ehrenwörtliche Erklärung**

---

### **Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB TU Darmstadt**

Hiermit versichere ich, Philip Aust, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

Darmstadt, den 29. Mai 2019

---



---

## Zusammenfassung

---

Durch die zunehmende Automatisierung von Fahrzeugen steigen die Anforderungen an die Absicherung der Fahrfunktionen. Die Durchführung von Simulationen bietet das Potential, die Freigabe hochautomatisierter Fahrfunktionen deutlich zu beschleunigen. Hierzu werden am Fachgebiet Fahrzeugtechnik (FZD) der Technischen Universität Darmstadt im Rahmen der Forschungsprojekte PE-GASUS und ENABLE-S3 Sensormodelle für Lidar-, Radar- und Ultraschallsensoren entwickelt.

Bisherige Simulationen arbeiten mit physikalisch basierten Sensormodellen, die die simulierte Umgebung erfassen und als Punktwolke ausgeben. Zur Erstellung einer Objektliste aus den Punktwolken sind anschließend aufwendige Algorithmen für Segmentierung, Tracking und Klassifizierung der Objekte erforderlich. Im Rahmen dieser Arbeit wird ein neuartiges Objektlisten-Modell für Lidar-Sensoren entwickelt, das Objektlisten generiert und ohne die Ausführung von rechenintensiven Algorithmen zur Punktwolkenverarbeitung auskommt. Dazu werden nur die Punktwolken des Lidar-Sensormodells sowie die bekannten Ground-Truth-Informationen aus der Simulationsumgebung verwendet.

Als Grundlage für das Objektlisten-Modell dient der Ibeo Lux 2010 sowie das mitgelieferte Steuergerät, das die Punktwolken zu Objektlisten verarbeitet. Zunächst wird ein Basismodell erstellt, das die real aufgezeichneten Objektlisten simuliert. Anschließend wird das Modell erweitert, sodass ebenfalls modernere Lidarsensoren mit verbesserten Algorithmen simuliert werden können.

Nach einer Einführung in die Grundlagen der Lidarsensorik wird das am Fachgebiet entwickelte Sensormodell sowie die nachfolgenden Algorithmen zur Bestimmung der Objektliste erläutert. Danach wird das Open Simulation Interface als generische Schnittstelle für das Modell vorgestellt, das eine modulare Anwendung des Objektlisten-Modells sicherstellt.

Anhand aufgezeichneter Messdaten des Ibeo Lux 2010 werden allgemeine sowie sensorspezifische Merkmale lidartypischer Objektlisten identifiziert und analysiert, die bei der Erstellung des Modells berücksichtigt werden sollen.

Anschließend wird die allgemeine Struktur des Objektlisten-Modells vorgestellt. Für die Implementierung des Modells werden verschiedene Ansätze zur Verwendung der Ground Truth und Punktwolken dargestellt, die die Verarbeitungsschritte Segmentierung, Klassifizierung und Tracking durchführen. Nach Diskussion und Bewertung der Ansätze wird die bestmögliche Lösung für das Modell des Ibeo Lux 2010 implementiert. Weiterhin wird das Modell dahingehend erweitert, dass ein moderner Lidar-Sensor mit verbesserter Bestimmung der Position, Geschwindigkeit und Abmessungen simuliert werden kann.

Die Qualität des entwickelten Sensormodells wird anhand von realen Fahrversuchen untersucht. Die Fahrversuche werden mithilfe der aufgezeichneten GPS-Daten nachsimuliert, sodass die Ausgabe des Objektlistenmodells direkt mit den realen Messdaten verglichen werden kann. Die Ergebnisse werden dabei durch Anwendung vorgestellter Metriken für Objektlisten bewertet. Im Anschluss wird ein Ausblick über die Möglichkeiten zur Erweiterung des Objektlisten-Modells gegeben.

---

## Inhaltsverzeichnis

---

Zusammenfassung .....	I
Inhaltsverzeichnis .....	II
Formelzeichen und Indizes .....	III
Abkürzungsverzeichnis .....	V
Abbildungsverzeichnis .....	VI
Tabellenverzeichnis .....	VIII
1 Einleitung .....	1
1.1 Motivation .....	1
1.2 Konkretisierung der Aufgabenstellung .....	1
1.3 Methodik des Vorgehens .....	2
2 Grundlagen .....	4
2.1 Lidarsensorik .....	4
2.2 Sensorsimulation am Fachgebiet FZD .....	9
2.3 Open Simulation Interface .....	15
3 Merkmale lidartypischer Objektlisten .....	17
3.1 Existenzunsicherheiten .....	17
3.2 Zustandsunsicherheiten .....	18
3.3 Klassenunsicherheiten .....	21
4 Implementierung des Modells .....	23
4.1 Modellstruktur .....	23
4.2 Ideale Objektliste .....	23
4.3 Modellierung des Ibeo Lux 2010 .....	24
4.4 Erweiterung des Ibeo-Modells .....	30
4.5 Übersicht über die verschiedenen Modellvarianten .....	35
5 Evaluation des Modells .....	37
5.1 Durchführung von Fahrversuchen .....	37
5.2 Simulation der Fahrversuche .....	39
5.3 Metriken .....	40
6 Ergebnisse .....	42
6.1 Vergleich von realen und simulierten Objektlisten .....	42
6.2 Vergleich der simulierten Objektlisten aus Ibeo-Modell und erweitertem Modell .....	50
7 Diskussion der Ergebnisse .....	57
7.1 Einfluss der Methodik .....	57
7.2 Einfluss der Simulationsumgebung .....	57
7.3 Einfluss des Sensormodells .....	58
8 Fazit und Ausblick .....	60
A Anhang .....	62

---

## Formelzeichen und Indizes

---

Lateinische Buchstaben:

Zeichen	Einheit	Bezeichnung
$b$	m	Fahrzeugbreite
$c$	$\frac{\text{m}}{\text{s}}$	Lichtgeschwindigkeit
$C$	m	Cut-Off-Parameter
$d$	m	Distanz, Entfernung
$f$	Hz	Frequenz
$h$	m	Fahrzeughöhe
$J$	-	Jaccard-Index
$k$	-	Korrekturfaktor
$l$	m	Fahrzeuglänge
$N$	-	Anzahl
$p$	-	Ordnung der OSPA-MT Metrik
$P$	m	Position
$Q$	-	Qualitätswert für die Bestimmung der Objektabmessungen
$r$	m	Suchradius
$t$	s	Zeit
$v$	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit
$x, x'$	m	Kartesische Koordinate
$y, y'$	m	Kartesische Koordinate
$z, z'$	m	Kartesische Koordinate

Griechische Buchstaben:

Zeichen	Einheit	Bezeichnung
$\alpha$	°, rad	Azimut
$\delta$	°, rad	Sensorauflösung
$\tau$	s	Zeitdauer
$\Phi$	W	Strahlungsleistung
$\varphi$	°, rad	Orientierung
$\omega, \omega'$	-	Teilmengen mit Trajektorien

---

Indizes:

<b>Zeichen</b>	<b>Bezeichnung</b>
a	absorbiert
b	Blickwinkel
B	Breite
abs	absolut
h	horizontal
L	Länge
max	maximal
PW	Punktewolke
r	reflektiert
rel	relativ
seg	segmentierte Punktewolke
t	transmittiert
v	vertikal
V	Verdeckung

---

## Abkürzungsverzeichnis

---

ADAS	Advanced Driver Assistance System
BB	Bounding Box
CNN	Convolutional Neural Network
FAS	Fahrerassistenzsystem
FZD	Fahrzeugtechnik Darmstadt
GT	Ground Truth
HiL	Hardware in the Loop
IoU	Intersection-over-Union
Lidar	Light Detection And Ranging
OSI	Open Simulation Interface
OSPA-MT	Optimal Subpattern Assignment for Multiple Tracks
Radar	Ratio Detection And Ranging
RBNN	Radially Bounded Nearest Neighbour
RMSE	Root Mean Squared Error
ROS	Robot Operating System
SiL	Software in the Loop
ToF	Time of Flight
ViL	Vehicle in the Loop
VTD	Virtual Test Drive

---

## Abbildungsverzeichnis

---

1-1	Struktur der Sensorsimulation.....	2
1-2	Methodik des Vorgehens .....	3
2-1	Time of Flight Prinzip .....	4
2-2	Pulsantwort zweier Objekte .....	5
2-3	Absorption, Reflexion, Transmission.....	5
2-4	Lambert-Reflektor, Transmission .....	6
2-5	Ausführungsformen von Lidarsensoren.....	6
2-6	Scanebenen des Ibeo Lux 2010 .....	7
2-7	Arbeitsbereich des Ibeo Lux 2010 .....	7
2-8	Variabe Winkelauflösung des Ibeo Lux 2010 .....	8
2-9	Konstante Winkelauflösung des Ibeo Lux 2010 .....	8
2-10	Position der Lidarsensoren im Versuchsfahrzeug.....	9
2-11	Ablauf der FZD-Sensorsimulation.....	10
2-12	Adaption des Suchradius für RBNN-Algorithmus .....	12
2-13	Struktur eines Tracking-Algorithmus .....	14
2-14	Komponenten der Sensorsimulation.....	15
2-15	Relevante Klassen des OSI-Interfaces.....	16
3-1	False Positives .....	18
3-2	Objektanmessungen abhängig von Position und Orientierung.....	18
3-3	Zustandsunsicherheiten am Rand des Sensor-Sichtfeldes .....	19
3-4	Wandernder Referenzpunkt .....	20
3-5	Fehlerhafte Positionsbestimmung .....	20
3-6	Verdeckung von Objekten .....	21
3-7	Beispiel für Klassenunsicherheiten .....	22
4-1	Struktur des Objektlisten-Sensormodells .....	23
4-2	Modellbasierte Segmentierung.....	24
4-3	Segmentierung mittels Ground Truth Informationen.....	25
4-4	Bestimmung der Bounding Box mit Ground Truth.....	26
4-5	Bestimmung der Bounding Box mithilfe der segmentierten Punktwolke .....	27
4-6	Methodik des implementierten Tracking-Algorithmus .....	28
4-7	Zuordnung von Objektlänge und -breite bei stehenden Objekten .....	30
4-8	Methodik der verbesserten Positionsbestimmung .....	31
4-9	Bestimmung eines Qualitätswerts abhängig vom Blickwinkel .....	33
4-10	Korrekturfaktor für die Objektentfernung .....	34
4-11	Bestimmung des Verdeckungsgrades eines Objektes .....	34
5-1	Kalibrierung der Lidarsensoren .....	37
5-2	Aufbau des ersten Szenarios.....	38
5-3	Aufbau des zweiten Szenarios.....	39

5-4	Versuch 1 in VTD .....	40
5-5	Schnittmenge und Vereinigungsmenge .....	41
6-1	Reale und simulierte Trajektorien, Versuch 1 .....	42
6-2	Unterhalb des Crafters verlaufende Lidarstrahlen .....	43
6-3	Reale und simulierte Relativgeschwindigkeiten, Versuch 1 .....	43
6-4	Reale und simulierte Objektabmessungen als Boxplots, Versuch 1 .....	44
6-5	OSPA-MT Metrik für reale Daten und Ibeo-Modell, Versuch 1 .....	46
6-6	OSPA-MT Metrik für relevante reale Daten und Ibeo-Modell, Versuch 1 .....	46
6-7	IoU für reale Daten und Ibeo-Modell, Versuch 1 .....	47
6-8	Reale und simulierte Trajektorien, Versuch 2 .....	47
6-9	Reale und simulierte Relativgeschwindigkeiten, Versuch 2 .....	48
6-10	Reale und simulierte Objektabmessungen als Boxplots, Versuch 2 .....	49
6-11	OSPA-MT Metrik für relevante reale Daten und Ibeo-Modell, Versuch 2 .....	49
6-12	IoU für reale Daten und Ibeo-Modell, Versuch 2 .....	50
6-13	Trajektorien von idealem Modell, Ibeo-Modell und verbessertem Modell, Versuch 2 .....	51
6-14	OSPA-MT-Metrik für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2 .....	51
6-15	IoU für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2 .....	52
6-16	Relativgeschwindigkeiten von erweitertem Modell, Versuch 2 .....	52
6-17	OSPA-MT-Metrik für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 2 .....	53
6-18	Objektabmessungen von Ibeo-Modell und verbessertem Modell, Versuch 2 .....	54
6-19	Objektabmessungen von Ibeo-Modell und verbessertem Modell als Boxplots, Versuch 2 .....	55
6-20	IoU für erweitertes Modell mit verbesserter Bestimmung der Objektabmessungen, Versuch 2 .....	56
A-1	Soll-Zeitplan .....	62
A-2	Ist-Zeitplan .....	63
A-3	OSPA-MT Metrik für reale Daten und Ibeo-Modell, Versuch 2 .....	64
A-4	Trajektorien von idealem Modell, Ibeo-Modell und verbessertem Modell, Versuch 1 .....	64
A-5	OSPA-MT-Metrik für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1 .....	65
A-6	IoU für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1 .....	65
A-7	Relativgeschwindigkeiten von erweitertem Modell, Versuch 1 .....	66
A-8	OSPA-MT-Metrik für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 1 .....	66
A-9	Objektabmessungen von verbessertem Modell, Versuch 1 .....	67
A-10	Objektabmessungen von verbessertem Modell als Boxplots, Versuch 1 .....	67
A-11	IoU für erweitertes Modell mit verbesserter Bestimmung der Objektabmessungen, Versuch 2 .....	68

---

## Tabellenverzeichnis

---

2-1	Einbauposition und -orientierung des Ibeo Lux 2010 im Versuchsfahrzeug vor der Kalibrierung.....	9
2-2	Klassifikationsmerkmale .....	15
3-1	Zusammensetzung des Zustands eines Objektes .....	17
4-1	Ausgabe der verschiedenen Modellvarianten .....	36
5-1	Kalibrierte Gierwinkel der Lidarsensoren .....	37
5-2	Verwendete Versuchsfahrzeuge und Äquivalente in VTD .....	40
6-1	RMSE für reale Daten und Ibeo-Modell, Versuch 1 .....	45
6-2	RMSE für reale Daten und Ibeo-Modell, Versuch 2 .....	48
6-3	RMSE für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2.....	51
6-4	RMSE für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 2 ....	53
6-5	RMSE für erweitertes Modell mit verbesserter Bestimmung der Objektabmessungen, Versuch 2.....	55
A-1	RMSE für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1.....	64
A-2	RMSE für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 1 ....	66
A-3	RMSE für erweitertes Modell mit verbesserter Bestimmung der Objektabmessungen, Versuch 1 .....	68



---

# 1 Einleitung

---

Autonomes Fahren (ab SAE Level 3<sup>1</sup>) ist einer der aktuellen Schwerpunkte der Forschung und Entwicklung in der Fahrzeugindustrie. Hochautomatisierte Fahrzeuge haben das Ziel, zunächst erhöhten Komfort für den Fahrer bei gleicher Sicherheit zu gewährleisten. Vollautomatisiertes Fahren soll dann in einem folgenden Schritt zu einer höheren Sicherheit als bei einem menschlichen Fahrer führen.

Nicht zuletzt durch die mediale Berichterstattung<sup>2</sup> steht die Sicherheit moderner Fahrerassistenzsysteme (FAS) im Fokus des öffentlichen Interesses. Daher ist die Absicherung autonomer Fahrfunktionen eine der Kernaufgaben der Automobilindustrie.<sup>3</sup>

## 1.1 Motivation

Ein wichtiger Maßstab für die Beurteilung der Sicherheit ist das Auftreten schwerwiegender Unfälle. In Deutschland passiert statistisch gesehen alle 210 Millionen gefahrene Kilometer ein fataler Unfall. Um nachzuweisen, dass ein autonom gesteuertes Fahrzeug mit einer statistischen Sicherheit von 95 % die Anzahl der Unfälle im Gegensatz zu einem menschlichen Fahrer halbiert, muss eine 10-fach längere Strecke und somit 2,1 Milliarden Kilometer zurückgelegt werden.<sup>4</sup> Dies ist nicht möglich, selbst die Fahrzeugflotte der Google-Tochter Waymo ist seit 2009 erst 10 Millionen Kilometer autonom gefahren.<sup>5</sup>

Eine mögliche Lösung stellt die Nutzung virtueller Simulationsumgebungen dar.<sup>6,7</sup> Die autonomen Fahrfunktionen können als Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL) oder Vehicle-in-the-Loop (ViL) getestet werden. Für repräsentative Simulationsergebnisse sind sehr genaue Modelle für die umgebungserfassenden Sensoren erforderlich. Im Rahmen dieser Masterthesis wird die Modellierung eines Lidar-Sensors betrachtet. Das Fachgebiet Fahrzeugtechnik (FZD) der TU Darmstadt beteiligt sich an den Projekten PEGASUS<sup>8</sup> und ENABLE-S3<sup>9</sup>, die das Ziel haben, einheitliche Standards für die Absicherung autonomer Fahrfunktionen herzuleiten.

## 1.2 Konkretisierung der Aufgabenstellung

In vorherigen Arbeiten am Fachgebiet FZD wurde bereits ein Lidar-Sensormodell sowie Algorithmen entwickelt, die die simulierten Punktwolken verarbeiten und eine Objektliste der Umgebung erstellen. Da in einer Simulation die Informationen über alle vorhandenen Objekte bekannt sind, bietet es

---

<sup>1</sup> SAE International: J3016 (2018)

<sup>2</sup> Zeit Online: Tesla-Unfall: Ermittler machen Autopilot für tödlichen Unfall mitverantwortlich (2017)

<sup>3</sup> Winner, H.: Quo vadis, FAS? (2015)

<sup>4</sup> Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2015)

<sup>5</sup> Waymo: Million of miles driven (2018)

<sup>6</sup> Sovani, S.: Simulation Accelerates Development of Autonomous Driving (2017)

<sup>7</sup> Winner, H.: How to Address the Approval Trap for Autonomous Vehicles (2015)

<sup>8</sup> Pegasus: Homepage Forschungsprojekt PEGASUS (2018)

<sup>9</sup> ENABLE-S3: Homepage Forschungsprojekt ENABLE-S3 (2018)

sich an, diese ideale Objektliste als Ground Truth (GT) Daten zu verwenden und ein objektbasiertes Sensormodell zu entwickeln.

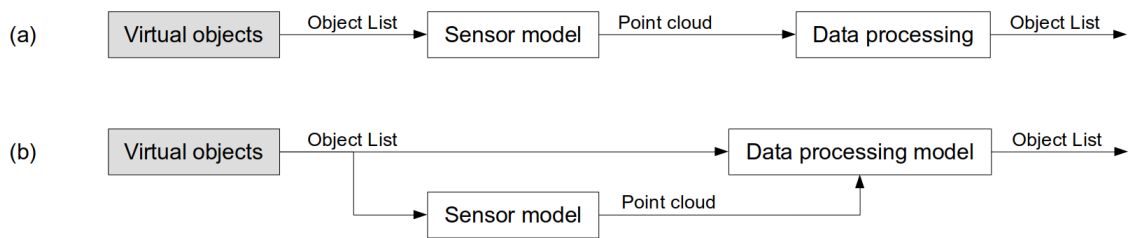


Abbildung 1-1: Struktur der Sensorsimulation. a: Konventionelle Sensorsimulation. b: Neuer Ansatz zur Sensorsimulation

Im Rahmen dieser Masterthesis soll ein solches Modell entstehen, welches lidartypische Objektlisten liefert, möglichst ohne die Algorithmen zur Objekt-Detektion, -Klassifikation und -Tracking durchzuführen. Abbildung 1-1 (b) zeigt die Struktur des neu zu entwickelnden Objektlisten-Modells. Im Gegensatz zur bisher verwendeten Struktur (a) wird zusätzlich zur Punktwolke des Lidar-Sensormodells auch die ideale Objektliste aus der Simulationsumgebung verwendet.

Im ersten Schritt der Arbeit soll eine Literaturrecherche über den Stand der Technik von Lidar-Sensoren und deren Simulation auf Punktwolken- und Objektlistenebene durchgeführt werden.

Anschließend sollen die möglichen Unterschiede zwischen idealen und lidartypischen Objektlistenmodellen herausgearbeitet werden.

Nach der Erarbeitung und Bewertung von verschiedenen Ansätzen zur Verwendung von Ground Truth und Punktwolken soll eine bestmögliche Kombination als Objektlisten-Sensormodell implementiert werden. Dies soll in einem definierten Framework unter Verwendung des Open Simulation Interface (OSI) geschehen.

Abschließend soll das Modell durch Heranziehen von Messdaten und den Vergleich mit realen Messungen evaluiert werden.

### 1.3 Methodik des Vorgehens

In Abbildung 1-2 ist die Methodik dargestellt, mit der bei dieser Masterthesis vorgegangen wird.

Zunächst wird eine Literaturrecherche zum Stand der Technik durchgeführt. Dabei werden die Grundlagen von Lidarsensoren, die Spezifikationen des verwendeten Sensors und das aktuell am Fachgebiet FZD verwendete Sensormodell sowie der Algorithmen zur Verarbeitung der Punktwolken betrachtet (s. Kapitel 2).

In Kapitel 3 werden Messungen des zugrunde liegenden Lidarsensors analysiert, um lidartypische Merkmale von Objektlisten in Form von Existenz-, Zustands- und Klassifizierungsunsicherheiten zu identifizieren.

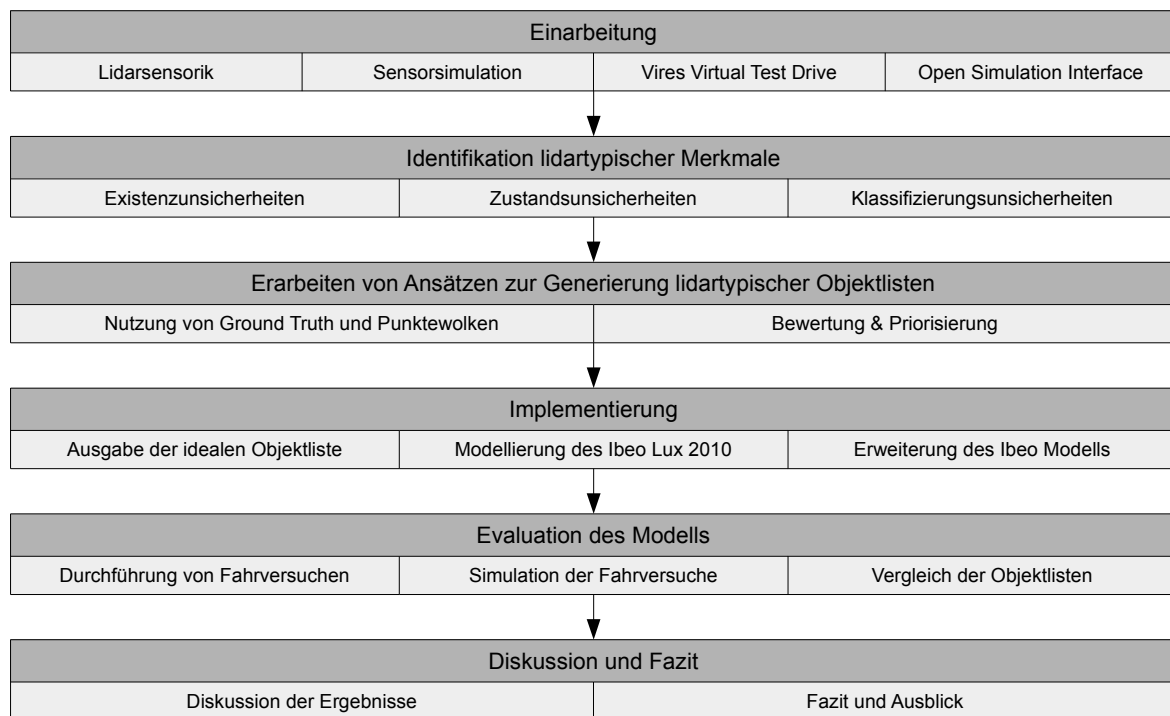


Abbildung 1-2: Methodik des Vorgehens

Anschließend werden Ansätze erarbeitet, um ein Objektlisten-Modell zu implementieren, das die zuvor identifizierten Merkmale unter Nutzung der Ground Truth und Punktwolken berücksichtigt. Die Ansätze werden bewertet und priorisiert.

Abschnitt 4 beschreibt die Implementierung des Modells. Zunächst wird dabei ein Modell implementiert, das die ideale Objektliste aus der Simulationsumgebung ausgibt. Anschließend wird ein Modell des Ibeo Lux 2010 erstellt, das die wichtigsten Unsicherheiten des Sensors berücksichtigt. Außerdem wird das Modell so erweitert, sodass auch Sensoren mit verbesserten Algorithmen simuliert werden können.

Die Vorgehensweise bei der Evaluation wird in Kapitel 5 vorgestellt. Dazu werden die durchgeführten Fahrversuche und deren Simulation beschrieben. Außerdem werden geeignete Metriken bestimmt, mit denen die realen und simulierten Objektlisten verglichen werden können.

Die Ergebnisse der angewandten Metriken werden in Kapitel 6 dargestellt. Anschließend werden die Resultate in Kapitel 7 kritisch diskutiert.

In Kapitel 8 werden die Ergebnisse dieser Arbeit zusammengefasst und ein Ausblick über zukünftige Arbeiten gegeben.

---

## 2 Grundlagen

---

In diesem Kapitel werden die Grundlagen von Lidar-Sensoren sowie deren Simulation auf Punktwolken- und Objektlistenebene erläutert. Zu Beginn wird der generelle Aufbau und das Funktionsprinzip des Lidar-Sensors erklärt sowie die Spezifikationen des in dieser Arbeit verwendeten Sensors vorgestellt. Anschließend wird auf die Simulation mithilfe von Sensormodellen, sowie das vorhandene Lidar-Modell und die Algorithmen von FZD eingegangen.

### 2.1 Lidarsensorik

LiDAR (Light Detection And Ranging) ist ein optisches Messverfahren zur Ortung und Messung der Entfernung von Objekten im Raum und verwendet Infrarot-, Ultraviolett- oder Strahlen im Bereich des sichtbaren Lichts.<sup>10a</sup>

#### 2.1.1 Aufbau und Funktionsprinzip

Das am meisten verwendete Messverfahren ist die „Time of Flight“-Methode. Dabei werden Lichtimpulse ausgesendet und an evtl. vorhandenen Objekten reflektiert. Das reflektierte Signal wird anschließend an einem Empfänger detektiert und die Zeitdauer zwischen Aussendung und Empfang gemessen (vgl. Abbildung 2-1). Dabei ist die Zeit proportional zur radialen Entfernung des Objekts. Mithilfe der Lichtgeschwindigkeit  $c$  und der gemessenen Zeitdauer  $\tau$  kann die Entfernung zwischen Sensor und Objekt berechnet werden:

$$d = \frac{c \cdot \tau}{2} \quad (2-1)$$

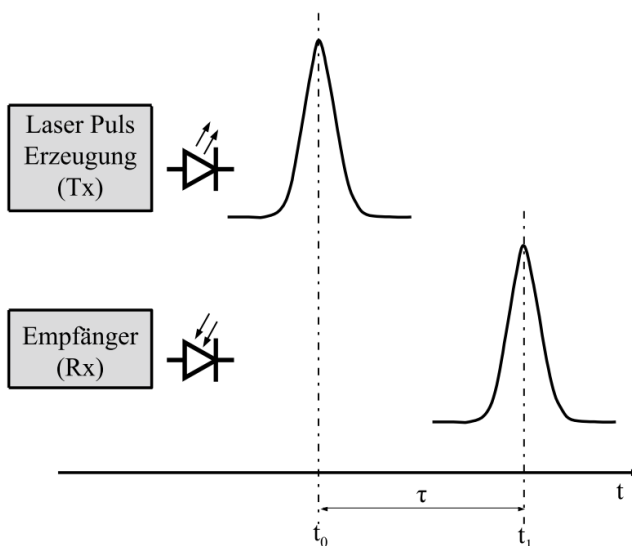


Abbildung 2-1: Time of Flight Prinzip.<sup>10b</sup>

---

<sup>10</sup> Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015) . a: S. 317-334; b: S. 319; c: S. 319; d: S. 324; e: S. 324; f: S. 325; g: S. 330.

Einige Lidarsensoren sind in der Lage, mehrere Echos pro ausgesendetem Strahl zu unterscheiden (vgl. Abbildung 2-2).

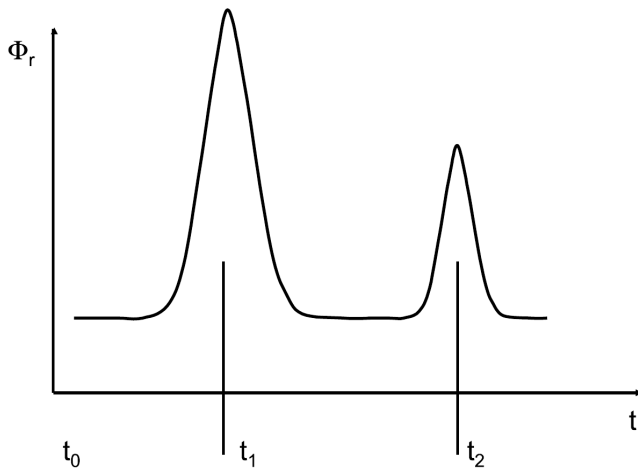


Abbildung 2-2: Pulsantwort zweier Objekte.<sup>10c</sup>

Die maximal erreichbare Reichweite des Sensors ist von der ausgesendeten Lichtintensität, der Empfindlichkeit des Empfängers, der Transmission der Atmosphäre sowie der Größe und Reflektanz des Objekts abhängig. Ein Teil der ausgestrahlten Lichtleistung  $\Phi_0$  wird durch die Atmosphäre diffus reflektiert ( $\Phi_r$ ) sowie absorbiert ( $\Phi_a$ ), am Ende der Strecke steht nur noch die Lichtleistung  $\Phi_t$  zur Verfügung (vgl. Abbildung 2-3). Dabei muss das Licht die Strecke nach der Reflexion erneut in umgekehrter Richtung durchlaufen. Die Herausforderung besteht darin, die wegen der Augensicherheit limitierte Energie nach der Reflexion wieder zu empfangen. Objekte reflektieren die Strahlen wie bei einem Lambert-Reflektor nicht gerichtet, sondern inhomogen im Raumwinkel (vgl. Abbildung 2-4). Somit beträgt die Energie, die am Empfänger genutzt werden kann, maximal 20% der am Objekt reflektierten Energie.<sup>10d</sup>

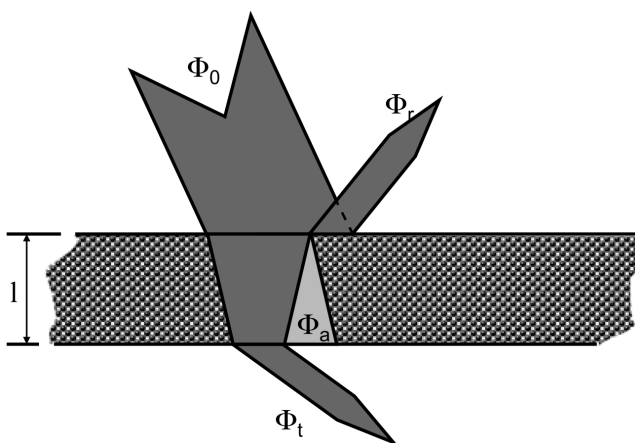


Abbildung 2-3: Absorption, Reflexion, Transmission.<sup>10e</sup>

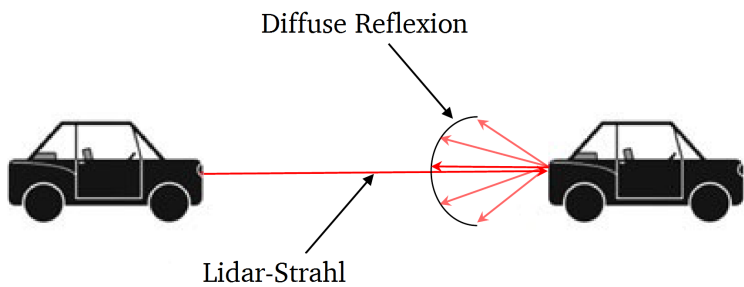


Abbildung 2-4: Lambert-Reflektor.<sup>10f</sup>

Abbildung 2-5 zeigt verschiedene Ausführungsvarianten von Lidarsensoren im Automobilbereich. Die Beispiele (a) und (c) nutzen einen oder mehrere starre Multibeam-Sensoren, um jederzeit den gesamten Erfassungsbereich zu überwachen. Variante (b) verwendet einen Multibeam-Sensor, der situationsabhängig ausgerichtet werden kann und somit immer nur einen Teil des Erfassungsbereich wahrnimmt. Bei Ausführungsform (d) handelt es sich um einen Scanner, der nacheinander alle Winkelsegmente des Erfassungsbereichs mit einem schmalen Strahl abtastet.

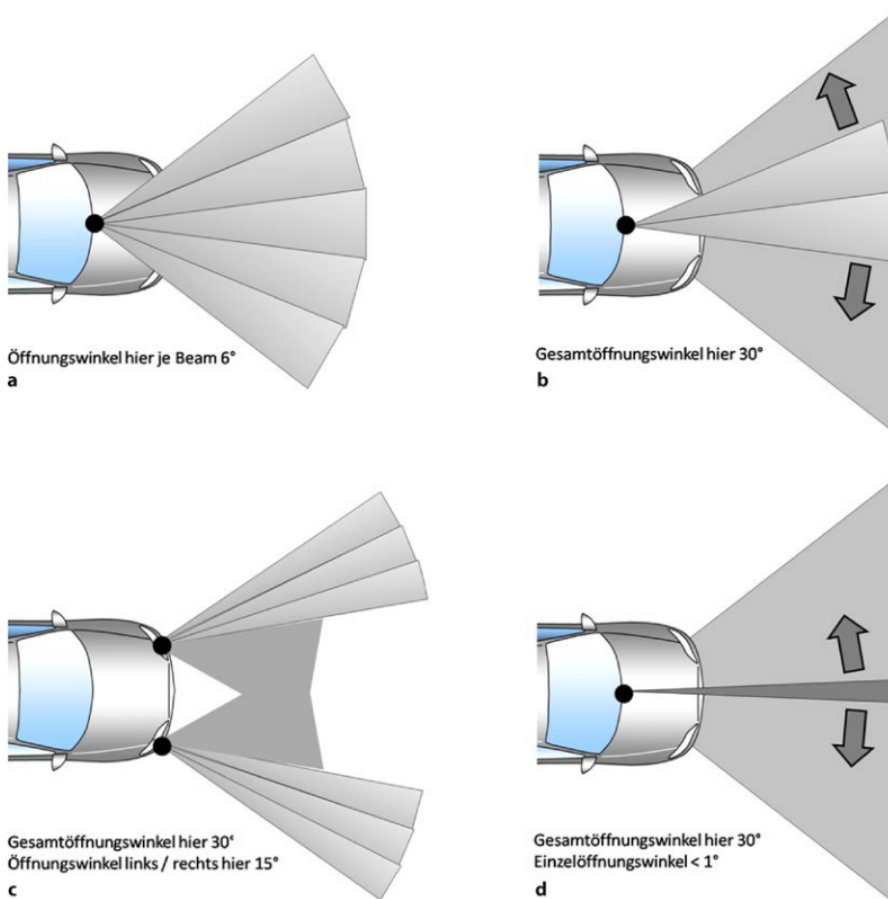


Abbildung 2-5: Ausführungsformen von Lidarsensoren.<sup>10g</sup>

### 2.1.2 Spezifikationen des ibeo Lux 2010

Im Versuchsfahrzeug und in der Simulation wird der Ibeo Lux 2010 eingesetzt. Der Sensor besitzt vier vertikale Scanebenen mit einem Öffnungswinkel von jeweils  $0,8^\circ$ , somit beträgt der vertikale Öffnungswinkel insgesamt  $3,2^\circ$  (vgl. Abbildung 2-6).<sup>11a</sup>

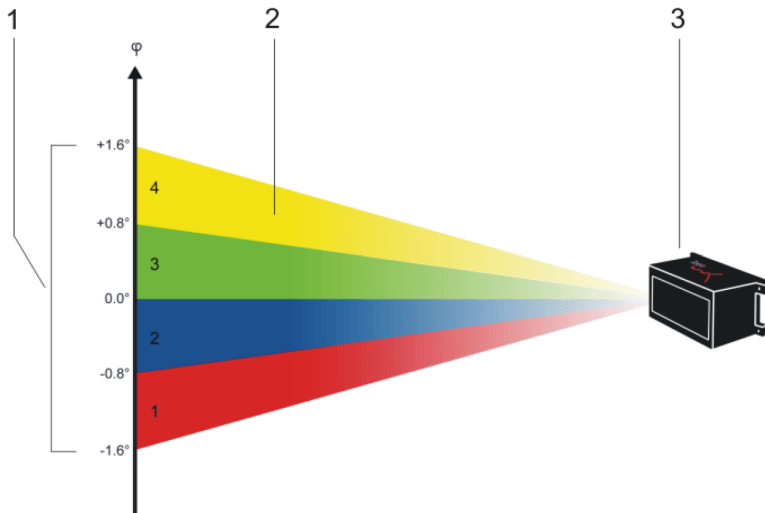


Abbildung 2-6: Scanebenen des Ibeo Lux 2010.<sup>11b</sup>

Der zentrale horizontale Arbeitsbereich (1) mit allen vier Scanebenen beträgt  $85^\circ$  (vgl. Abbildung 2-7). Der Arbeitsbereich kann auf  $110^\circ$  erweitert werden, wobei die äußeren Bereiche (2) dann nur noch mit zwei Scanebenen abgedeckt werden.

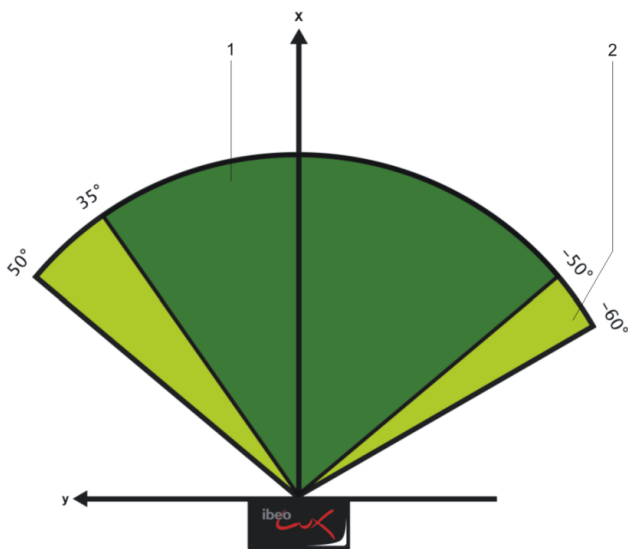


Abbildung 2-7: Arbeitsbereich des Ibeo Lux 2010.<sup>11c</sup>

Der Sensor kann mit drei verschiedenen Scanfrequenzen (12,5 Hz, 25 Hz und 50 Hz) betrieben werden. Bei einer Scanfrequenz von 12,5 Hz kann die Winkelauflösung in den Sektoren unterschiedlich

<sup>11</sup> Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® Laserscanner (2015) . a: S. 3-5; b: S. 3-5; c: S. 3-12; d: S. 3-8; e: S. 3-13; f: S. 9-2.

eingestellt werden. Der zentrale Bereich (1) bei  $\pm 10^\circ$  in Abbildung 2-8 ist beispielsweise für ACC relevant und besitzt eine Auflösung von  $0,125^\circ$ , um auch in größerer Entfernung noch mehrere Messwerte pro Objekt zu erhalten. Der mittlere Bereich (2) zwischen  $\pm 10^\circ \dots 30^\circ$  hat eine Auflösung von  $0,25^\circ$ , die Auflösung im äußeren Bereich (3) beträgt  $0,5^\circ$ .

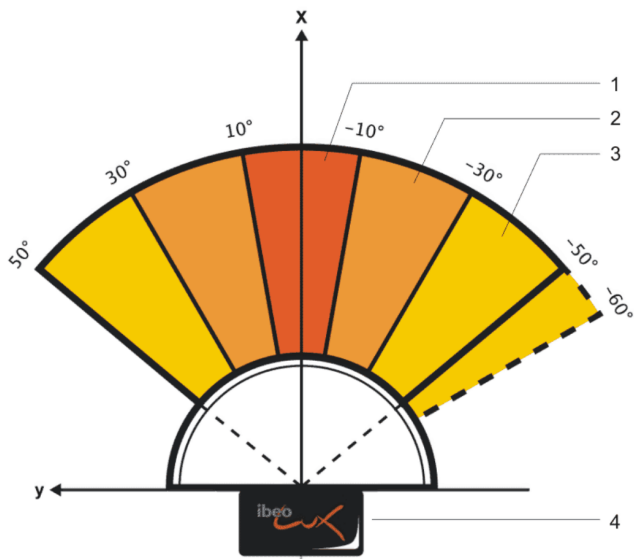


Abbildung 2-8: Variabe Winkelauflösung des Ibeo Lux 2010.<sup>11d</sup>

Im Versuchsfahrzeug wird der Sensor mit einer konstanten Auflösung von  $0,25^\circ$  bei einer Scanfrequenz von 12,5 Hz betrieben. Zusätzlich zur Verschiebung für einen größeren horizontalen Arbeitsbereich sind die Scanebenen noch um  $0,125^\circ$  zueinander verschoben, um eine höhere horizontale Auflösung zu erreichen (vgl. Abbildung 2-9).

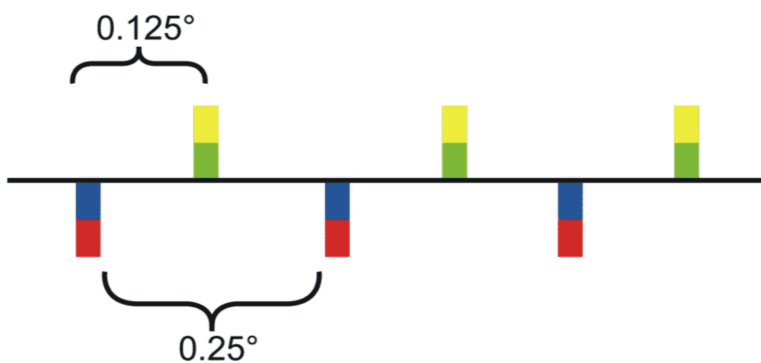


Abbildung 2-9: Konstante Winkelauflösung des Ibeo Lux 2010.<sup>11e</sup>

Der Messbereich wird mit 0,3 m bis 200 m in  $0^\circ$  Richtung angegeben, die Entfernungsauflösung beträgt 0,04 m. Die Pulsdauer beträgt ca. 4,5 ns. Pro ausgesendetem Lichtimpuls können bis zu drei Echos detektiert werden.

Im verwendeten Versuchsfahrzeug sind zwei ibeo Lux 2010 verbaut, die sich links und rechts in der vorderen Stoßstange befinden (vgl. Abbildung 2-10). Die beiden Punktwolken in Polarkoordinaten werden in einem Steuergerät fusioniert und im kartesischen Koordinatensystem, bezogen auf die Mitte



der Hinterachse des Fahrzeugs, ausgegeben. Anschließend werden Bodenreflexionen detektiert und die Punktwolke segmentiert, indem Gruppen von Punkten in einzelne Segmente zusammengefasst werden. Die Segmente werden beim sogenannten Tracking zeitlich verfolgt, zum Schluss werden die Segmente noch anhand der Objekteigenschaften klassifiziert.

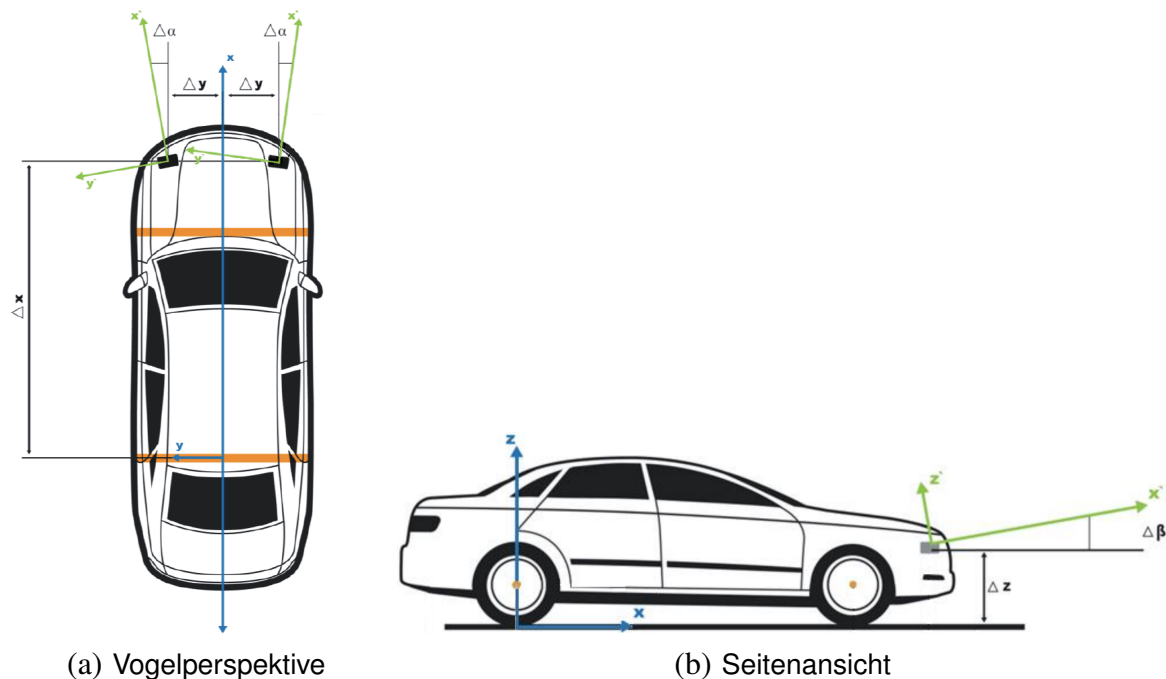


Abbildung 2-10: Position der Sensoren im Versuchsfahrzeug.<sup>11f</sup>

In Tabelle 2-1 ist die Einbauposition und -orientierung der Lidarsensoren im Versuchsfahrzeug angegeben. Vor der Durchführung der Fahrversuche wird der genaue Gierwinkel der Sensoren überprüft und kalibriert (siehe Kapitel 5).

Tabelle 2-1: Einbauposition und -orientierung des Ibeo Lux 2010 im Versuchsfahrzeug vor der Kalibrierung

	Versatz X	Versatz Y	Versatz Z	Nickwinkel	Wankwinkel	Gierwinkel
<b>Lidar 1</b>	3,47 m	0,63 m	0,36 m	0°	0°	17,906°
<b>Lidar 2</b>	3,49 m	-0,58 m	0,36 m	0°	0°	-17,906°

## 2.2 Sensorsimulation am Fachgebiet FZD

In Abbildung 2-11 ist der derzeitige Ablauf der Sensorsimulation am Fachgebiet FZD dargestellt. Als virtuelle Simulationsumgebung wird Virtual Test Drive (VTD) der Vires Simulationstechnik GmbH verwendet, in der statische (Straßen, Gebäude) und dynamische Objekte (Fahrzeuge, Fußgänger) simuliert werden können. Dabei kann ein Ego-Fahrzeug mit verschiedenen Sensoren ausgestattet werden, um die simulierte Umwelt zu erfassen. Am Fachgebiet FZD wurde bereits ein Modell

des Flughafengeländes in Griesheim, das für Testfahrten genutzt werden kann, entwickelt.<sup>12</sup> Dieses kann genutzt werden, um durchgeführte Fahrversuche mithilfe aufgezeichneter GPS-Daten in VTD nachzusimulieren und die gesammelten Messdaten miteinander zu vergleichen. Wie am realen Versuchsfahrzeug sind am Ego-Fahrzeug zwei Lidarsensoren angebracht, die ein Sensormodell des Ibeo Lux 2010 enthalten.

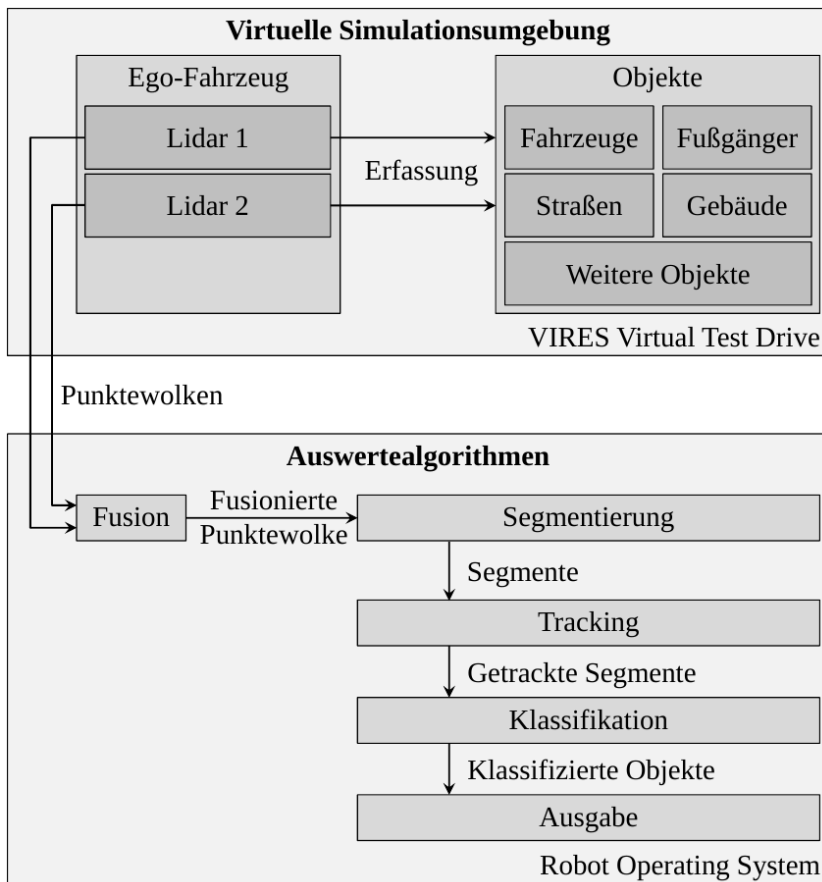


Abbildung 2-11: Ablauf der FZD-Sensorsimulation<sup>13a</sup>

Um eine Objektliste mit allen detektierten Objekten und deren Zuständen zu erhalten, müssen die simulierten Messdaten mit einer Reihe von Algorithmen ausgewertet werden. Dies wird im Software-Framework Robot Operating System (ROS) durchgeführt. Zuerst werden die Messdaten der zwei Sensoren zu einer gemeinsamen Punktwolke fusioniert. Anschließend werden Segmentierung, Tracking und Klassifizierung durchgeführt, die in den folgenden Abschnitten genauer betrachtet werden.

<sup>12</sup> Vijayendra, V. B.: Refinement of a Virtual Environment Representation to conduct validation tests for Automated driving (2018)

<sup>13</sup> Huch, S.: Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation (2018) . a: S. 12; b: S. 8.

---

### 2.2.1 Sensormodell

Das Sensormodell basiert auf der Methode des Ray-Casting, einer vereinfachten Form des Ray-Tracing.<sup>14</sup> Dabei werden ähnlich dem realen Sensorprinzip einzelne Lichtstrahlen simuliert. Treffen die simulierten Strahlen auf Objekte, werden die getroffenen Punkte in einer dreidimensionalen Punktwolke ausgegeben. Im Gegensatz zum Ray-Tracing wird die Reflexion der Strahlen, die abhängig von den Oberflächeneigenschaften eines Objektes ist, nicht berücksichtigt. Jin<sup>15</sup> hat in einer Masterarbeit das Modell so erweitert, dass der simulierten Punktwolke Rauscheffekte hinzugefügt werden können.

### 2.2.2 Segmentierung

Segmentierung bezeichnet die Unterteilung von Punktwolken in zusammengehörende Segmente. Dabei sollen Teilmengen der Punktwolke gefunden werden, die einem einzelnen Objekt (unabhängig von der Klasse) zugeordnet werden können. Außerdem beinhaltet die Segmentierung die Bodendetektion. Ein Herausfiltern der Punkte, die zum Boden gehören, erleichtert die Segmentierung der restlichen Objekte erheblich.

Nach Nguyen<sup>16</sup> kann die Segmentierung in fünf Kategorien eingeteilt werden: Kantenbasierte, regionsbasierte, attributsbasierte, modellbasierte sowie graphenbasierte Segmentierung.

Bei der kantenbasierten Segmentierung werden Kanten detektiert, um die Objekte voneinander abzugrenzen. Dazu werden Punkte gesucht, zwischen denen sich die Signalintensität sprunghaft ändert. Kantenbasierte Algorithmen sind sehr schnell, aber sehr störungsempfindlich gegenüber Rauschen und unterschiedlicher Punktwolkedichten.

Regionsbasierte Methoden verwenden Informationen über die direkten Nachbarn, um Punkte mit ähnlichen Eigenschaften zu gruppieren. Diese Methode ist robuster gegenüber Rauschen, hat aber oft Probleme, die genaue Grenze zwischen Objekten zu bestimmen.

Bei der attributsbasierten Segmentierung werden in einem ersten Schritt bestimmte Merkmale berechnet, anschließend wird die Punktwolke anhand dieser Attribute analysiert. Diese Art der Segmentierung ist sehr robust und präzise, allerdings ist sie abhängig von den gewählten Merkmalen und der Dichte der Punktwolke.

Modellbasierte Algorithmen verwenden primitive geometrische Formen (Ebenen, Kugeln, Zylinder, Kegel), um die Punkte zu gruppieren. Wird eine solche Form identifiziert, kann sie einem Objekt zugeordnet werden. Objekte mit einer anderen Gestalt können dabei nicht erkannt werden.

---

<sup>14</sup> Schuler, K.; Becker, D.; Wiesbeck, W.: Extraction of Virtual Scattering Centers of Vehicles by Ray-Tracing Simulations (2008)

<sup>15</sup> Jin, G.: Enhancement of an ideal Lidar sensor model with identified physical effects (2018)

<sup>16</sup> Nguyen, A.; Le, B.: 3D Point Cloud Segmentation: A survey (2013)

Bei FZD wird ein RBNN-Algorithmus nach Klasing<sup>17</sup> verwendet. Dabei müssen nur zwei Parameter, die minimale Segmentgröße und der maximale Abstand zwischen den Punkten in einem Segment, festgelegt werden. Ein weiterer Vorteil ist der geringe Rechenaufwand, da nicht von jedem Punkt der Abstand zu den nächsten Nachbarn berechnet werden muss.

Ein fester Suchradius führt bei einem zu kleinen Suchradius zu einer Untersegmentierung im Fernbereich und bei einem zu großen Suchradius zu einer Übersegmentierung im Nahbereich des Sensors. Daher ist es sinnvoll, den Suchradius abhängig von der Entfernung des Reflexionspunktes zu wählen.<sup>18a</sup> Abbildung 2-12 zeigt das Prinzip der Suchradius-Adaption.

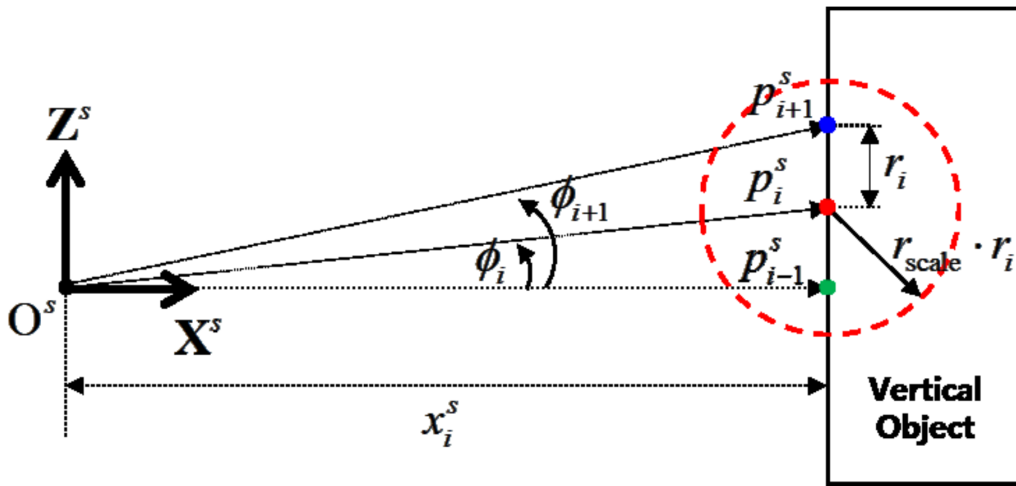


Abbildung 2-12: Adaption des Suchradius für RBNN-Algorithmus.<sup>18b</sup>

Beim Ibeo Lux 2010 ergibt sich für den Suchradius  $r_i$  mit dem vertikalen Öffnungswinkel  $\delta_v = 0,8^\circ$  zwischen den vier Scanebenen

$$r_i \approx x_i \cdot \tan(0,8^\circ). \quad (2-2)$$

Der Suchradius wird anschließend mit einem Skalierungsfaktor  $r_{scale}$  multipliziert, da die Reflexionspunkte eines Objekts nicht zwangsweise auf einer vertikalen Ebene liegen. Nach Knerr<sup>19a</sup> wird  $r_{scale} = 3$  gewählt, sodass sich für den Suchradius

$$r \approx 3 \cdot x_i \cdot \tan(0,8^\circ) \quad (2-3)$$

ergibt.

<sup>17</sup> Klasing, K.; Wollherr, D.; Buss, M.: A Clustering Method for Efficient Segmentation of 3D Laser Data (2008)

<sup>18</sup> Yungeun, C.; Seunguk, A.; Myung, J. C.: Fast Point Cloud Segmentation for an Intelligent Vehicle Using sweeping 2D laser scanners (2012) . a: S. ; b: S. 3.

<sup>19</sup> Knerr, J.: Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018)

---

### 2.2.3 Tracking

Tracking bezeichnet die zeitliche Verfolgung dynamischer Objekte in Echtzeit. Dazu wird die Anzahl bewegter Objekte im Messbereich und deren Zustände aus vorhergehenden und aktuellen Messungen bestimmt. Tracking-Algorithmen können nach Granström<sup>20</sup> in vier Gruppen unterteilt werden: Point Object Tracking, Extended Object Tracking, Group Object Tracking und Tracking With Multi-Path Propagation.

Beim Point Object Tracking generiert jedes Objekt pro Zeitschritt genau einen Messwert, d.h. ein Objekt belegt eine einzelne Auflösungszelle. Beim Extended Object Tracking generiert ein Objekt pro Zeitschritt mehrere Messwerte, die räumlich entlang des Objekts verteilt sind, sodass ein Objekt mehrere Auflösungszellen belegt. Beim Group Object Tracking werden ebenfalls mehrere Messwerte pro Zeitschritt generiert. Wenn mehrere Objekte die gleiche Bewegung durchführen, werden sie als Gruppe zusammengefasst und als Gesamtobjekt verfolgt. Tracking With Multi-Path Propagation berücksichtigt die Mehrwegausbreitung, es werden mehrere Messwerte pro Zeitschritt generiert, allerdings sind die Messwerte nicht mehr räumlich um das Objekt strukturiert.

In Abbildung 2-13 ist die allgemeine Struktur eines Tracking-Algorithmus dargestellt. In einem ersten Schritt werden die Sensordaten vorverarbeitet, Bodenreflexionen werden entfernt und die Punktwolke segmentiert. Anschließend werden die neuen Messpunkte den existierenden Objekthypothesen zugeordnet. Wenn für einen Messwert noch keine Objekthypothese besteht, wird eine neue erstellt. Im Track-Management werden die Zustände aller Objekte verwaltet. Im bestehenden Extended-Object-Tracker von Knerr<sup>19b</sup> wird ein Objekt durch die Position seines Schwerpunktes ( $x_k$ ,  $y_k$  und  $z_k$ ), der Relativ- und Absolutgeschwindigkeit ( $v_{\text{rel},k}$  bzw.  $v_{\text{abs},k}$ ), der Orientierung ( $\varphi_k$ ) und die geometrischen Abmessungen ( $l_k$ ,  $w_k$  und  $h_k$ ) repräsentiert:

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & z_k & v_{\text{rel},k} & v_{\text{abs},k} & \varphi_k & l_k & w_k & h_k \end{bmatrix}^T \quad (2-4)$$

Jedem Objekt wird eine eindeutige ID zugeordnet. Die Objektliste wird genutzt, um Objekthypothesen für den nächsten Messzyklus zu prädictieren, welche mit den gemessenen Daten verglichen werden können. Der Zustand des Objekts wird im Track-Management mit einem Kalman-Filter aktualisiert. Wenn ein Objekt in fünf aufeinanderfolgenden Messungen detektiert wird, wird es initialisiert, wenn es in zehn aufeinanderfolgenden Messungen nicht mehr erkannt wird, wird es aus dem Track-Management gelöscht.

---

<sup>20</sup> Granström, K.; Baum, M.; Reuter, S.: Extended Object Tracking: Introduction, Overview and Applications (2017)

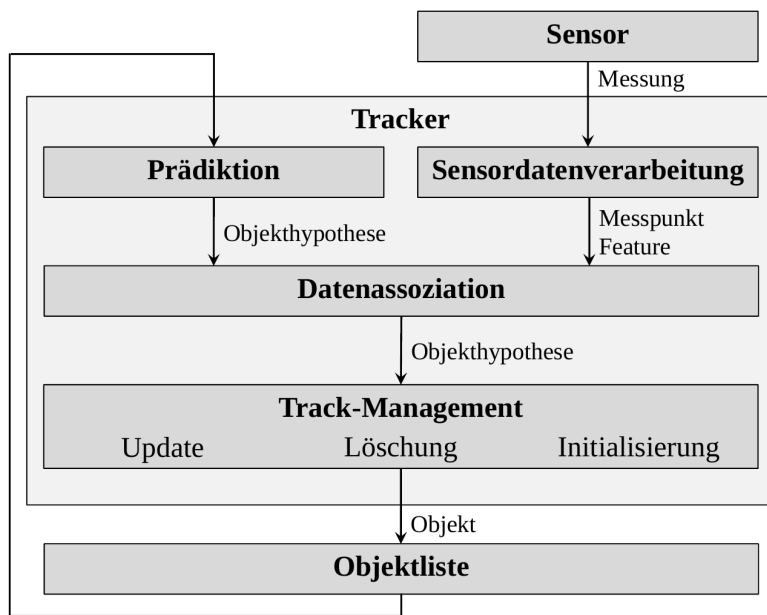


Abbildung 2-13: Struktur eines Tracking-Algorithmus.<sup>13b</sup>

In einer darauf aufbauenden Arbeit bei FZD von Zhang<sup>21</sup> wurde der Tracking-Algorithmus mit einem Unscented Kalman-Filter (UKF) erweitert.

#### 2.2.4 Klassifizierung

Die Klassifizierung hat das Ziel, allen detektierten Objekten eine spezifische Klasse zuzuordnen. Für das bestehende Sensormodell wurden von Ngo<sup>22</sup> zwei verschiedene Algorithmen zur Klassifizierung implementiert.

Der erste Algorithmus arbeitet mit einem End-to-end-Ansatz und kann Objekte direkt aus den unverarbeiteten Punktwolken klassifizieren. Die Basis beim sogenannten Deep Learning bilden Convolutional Neural Networks (CNN) aus dem Bereich des maschinellen Lernens. Für das FZD-Sensormodell wurde das generische Netzwerk VoxelNet<sup>23</sup> verwendet. Die Ergebnisse haben gezeigt, dass es in der Simulation des bei FZD verwendeten Ibeo Lux 2010 nicht geeignet ist, da dieser nur vier Kanäle hat und zu wenige Informationen über die Form der Objekte generiert.

Bei der mehrstufigen Objekterkennung werden die Objekte nicht direkt mit den Rohdaten des Sensors klassifiziert, die Sensordaten müssen zunächst wie in Abschnitt 2.2.2 beschrieben segmentiert werden. Anschließend wird aus den Segmenten eine Reihe von Merkmalen (vgl. Tabelle 2-2) bestimmt. Anhand dieser Merkmale kann die Klasse des jeweiligen Objekts bestimmt werden. In vorherigen

<sup>21</sup> Zhang, Y.: Implementation of an enhanced Lidar Tracking algorithm for automated driving (2018)

<sup>22</sup> Ngo, A.: Erweiterung eines LiDAR Sensormodells für simulationsbasiertes Entwickeln und Testen von Objekterkennungsalgorithmen (2018)

<sup>23</sup> Zhou, Y.; Tuzel, O.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection (2017)

Tabelle 2-2: Klassifikationsmerkmale

Merkmal	Beschreibung
length	Naive Länge, X-Richtung
width	Naive Breite, Y-Richtung
height	Naive Höhe, Z-Richtung
meanX	Mittlere Entfernung in X-Richtung
meanY	Mittlere Entfernung in Y-Richtung
meanZ	Mittlere Entfernung in Z-Richtung
stdX	Standardabweichung der X-Werte
stdY	Standardabweichung der Y-Werte
stdZ	Standardabweichung der Z-Werte
distance	Distanz zum Mittelpunkt
np	Anzahl der Punkte
np/distance	Anzahl der Punkte pro Distanz
np · distance	Anzahl der Punkte multipliziert mit der Distanz

Arbeiten bei FZD<sup>24,25</sup> hat sich gezeigt, dass einige Merkmale (grau markiert) vernachlässigt werden können, ohne dass sich die Klassifizierung merklich verschlechtert.

Die eigentliche Klassenzuordnung geschieht mittels eines Entscheidungsbaums, dazu wurde ein Bagged Decision Tree mit 100 Entscheidungsbäumen und maximal 20 Abzweigungen pro Baum implementiert.

## 2.3 Open Simulation Interface

Das Open Simulation Interface (OSI) ist ein generisches Interface für die Simulation der Umgebungswahrnehmung von Funktionen zum automatisierten Fahren.<sup>26</sup> Dabei sind die Schnittstellen zwischen Simulationsumgebung und Entwicklungsframework so standardisiert, dass die Modularität der einzelnen Komponenten sichergestellt ist (vgl. Abbildung 2-14).

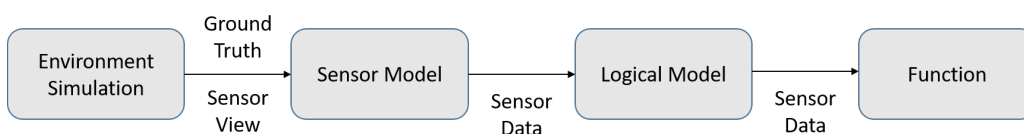


Abbildung 2-14: Komponenten der Sensorsimulation.<sup>27</sup>

Abbildung 2-15 zeigt die für das Sensormodell relevanten Klassen des OSI-Interfaces. Die Basis bildet die Struktur `osi3::SensorData`, in der alle GT Informationen sowie von Sensorenmodellen ab-

<sup>24</sup> Bert, F.: Identifikation von Merkmalen für die Objekterkennung mit lernenden LiDAR-Algorithmen (2017)

<sup>25</sup> Stäcker, L.: Entwicklung eines Informationssystems zum Segmentieren und Tracking von Objekten auf Basis von LiDAR-Sensordaten für Autonomes Fahren (2017)

<sup>26</sup> Hanke, T. et al.: Open Simulation Interface (2017)

<sup>27</sup> Friedmann, L.: Open Simulation Interface Wiki (2017)

geleitete Daten gespeichert werden. In *osi3::SensorView* werden die GT Informationen übergeben, *osi3::MovingObject* enthält Informationen von jedem dynamischen Objekt in der Simulation: Die Klasse, die Größe der Bounding Box (BB), die Position, Orientierung und Geschwindigkeit des Objektes im globalen Koordinatensystem. Äquivalent dazu gibt es ebenfalls eine Struktur für statische Objekte, das entwickelte Objektlistenmodell beschränkt sich aber zunächst nur auf dynamische Objekte. In *osi3::FeatureData* können Daten verschiedener Sensoren - in dieser Arbeit ausschließlich Lidar-Daten - gespeichert werden. Dabei wird jeder Punkt der dreidimensionalen Punktwolke mit seiner Position in sphärischen Koordinaten sowie weiteren Informationen wie Intensität oder Existenzwahrscheinlichkeit beschrieben. Dies wird genutzt, um die generierten Punktwolken der Sensormodelle zu übergeben. *osi3::DetectedMovingObject* wird verwendet, um die Ergebnisse der Algorithmen zur Objekterkennung als Objektliste abzuspeichern. Jedes detektierte dynamische Objekt wird wie die GT Informationen eines *osi3::MovingObject* mit Klasse, Positions- und Bewegungsparametern abgespeichert. Zusätzlich kann eine Existenzwahrscheinlichkeit, ein Referenzpunkt oder Bewegungszustand, etwa ob sich das Objekt gerade bewegt oder steht, gespeichert werden.

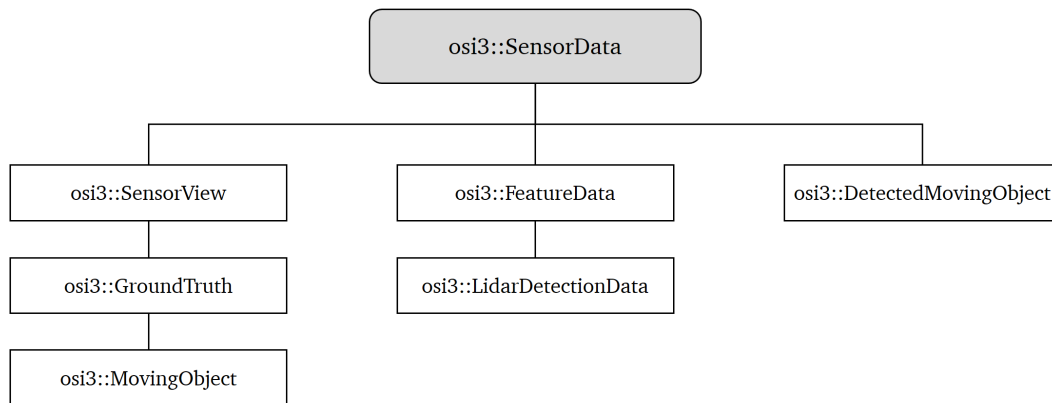


Abbildung 2-15: Relevante Klassen des OSI-Interfaces



---

### 3 Merkmale lidartypischer Objektlisten

---

Beim Tracking werden wahrgenommene Objekte in einer dynamischen Liste registriert und verwaltet. Dabei erhält jedes Objekt eine eindeutige ID, außerdem wird den Objekten eine Reihe von Eigenschaften zugeordnet. Tabelle 3-1 zeigt die wichtigsten Merkmale eines Objektes in einer Objektliste.

Tabelle 3-1: Zusammensetzung des Zustands eines Objektes

Merkmal	Beschreibung
ID	Eindeutiger Identifikator für das Objekt
Klasse	Semantische Bedeutung des Objektes
Abmessungen	Länge, Breite und Höhe der Bounding Box
Position	Relative Entfernung in X-, Y- und Z-Richtung
Geschwindigkeit	Relative Geschwindigkeit in X-, Y- und Z-Richtung
Beschleunigung	Relative Beschleunigung in X-, Y- und Z-Richtung
Orientierung	Gier-, Nick- und Wankwinkel
Orientierungsrate	Änderungsrate des Gier-, Nick- und Wankwinkels

Reale Objektlisten, die aus Lidarmessungen erzeugt werden, unterscheiden sich von idealen Objektlisten. Nach Dietmeier<sup>28</sup> existieren drei verschiedene Arten von Unsicherheiten: Existenz-, Zustands- und Klassenunsicherheiten. Im Folgenden werden die Unterschiede zwischen idealen und lidartypischen Objektlisten anhand von Messdaten des Ibeo Lux 2010 analysiert und in die jeweilige Kategorie eingeteilt.

#### 3.1 Existenzunsicherheiten

Die erste Kategorie behandelt Existenzunsicherheiten, die die Unsicherheit beschreiben, ob ein von der Sensorik entdecktes Objekt real existiert. Bei der Beurteilung der Existenzunsicherheiten können vier Fälle auftreten:

- True Positive (TP): Vorhandenes Objekt wurde erkannt
- False Negative (FN): Vorhandenes Objekt wurde nicht erkannt
- False Positive (FP): Nicht vorhandenes Objekt wurde fälschlicherweise erkannt
- True Negative (TN): Nicht vorhandenes Objekt wurde nicht erkannt

In Abbildung 3-1 ist ein Beispielfall dargestellt, bei dem am rechten Fahrbahnrand ein Fahrzeug korrekt erkannt wird (TP). Neben der Straße werden zwei Gruppierungen von Punkten fälschlicherweise als Objekt interpretiert (FP).

---

<sup>28</sup> Dietmeier, K.: Predicting of Machine Perception for Automated Driving (2016)

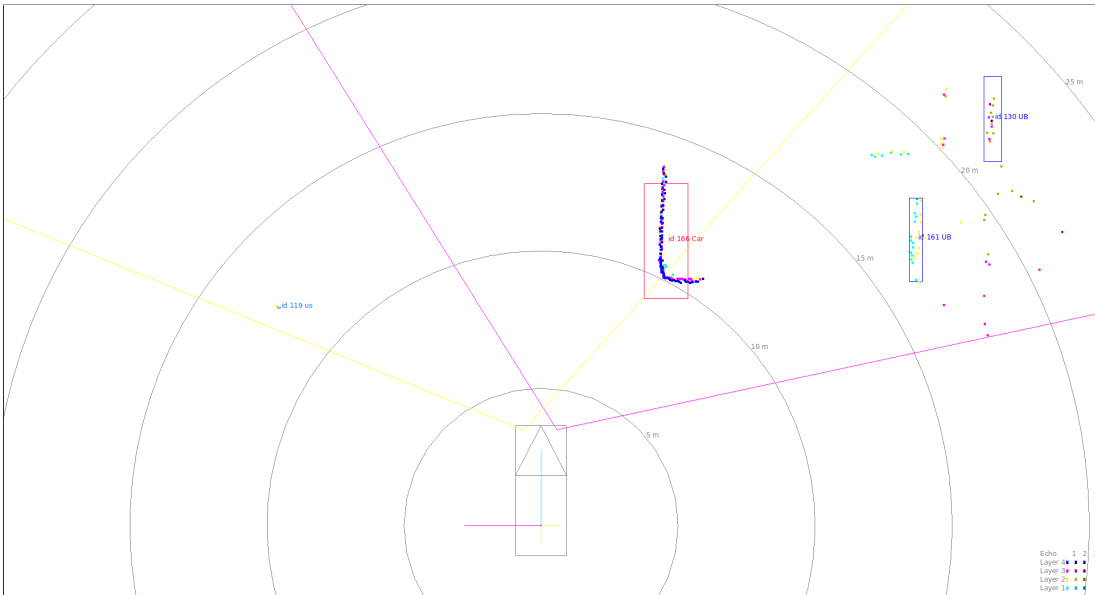


Abbildung 3-1: False Positives

### 3.2 Zustandsunsicherheiten

Die zweite Kategorie bilden die Zustandsunsicherheiten. Diese beschreiben die Unsicherheit bei der Bestimmung physikalischer Messgrößen wie Position, Geschwindigkeit, Abmessungen, Orientierung etc.

#### Abmessungen abhängig von Position und Orientierung

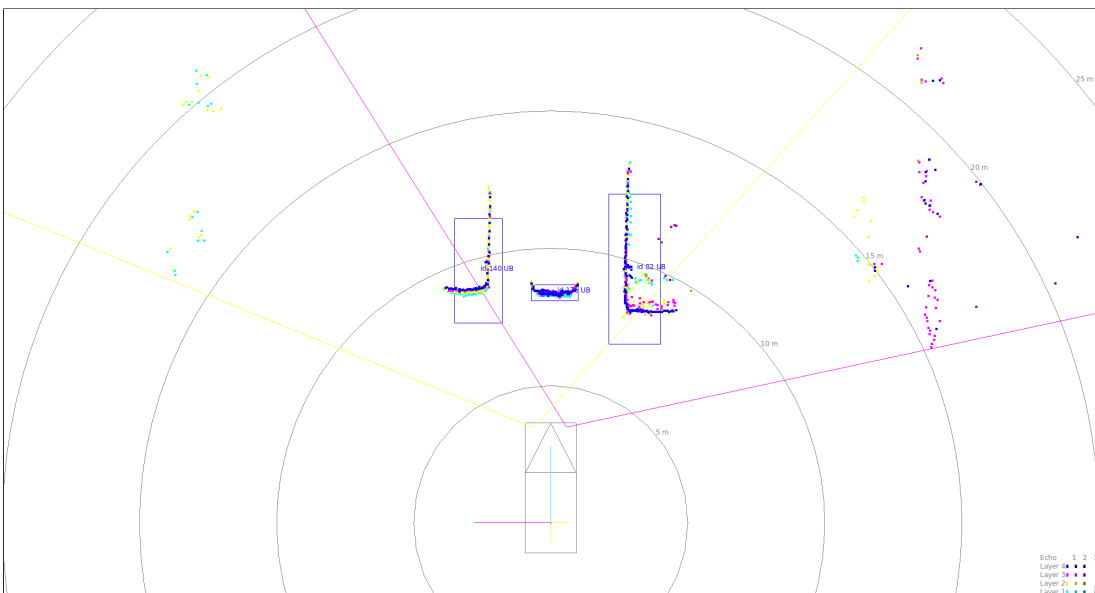


Abbildung 3-2: Objektanmessungen abhängig von Position und Orientierung

In Abbildung 3-2 ist eine Messung zu sehen, bei der sich das Versuchsfahrzeug frontal vor drei nebeneinander stehenden Fahrzeugen befindet. Beim zentralen Fahrzeug können die Lidarstrahlen nur

die Front erreichen, daher sind fast keine Informationen über die Länge des Fahrzeugs vorhanden. Dadurch wird die Bounding Box kleiner abgebildet als bei einer idealen Objektliste.

### Zuordnung der Länge und Breite eines Objektes

Aus der Form der segmentierten Punktwolke lassen sich Informationen über die Orientierung eines Objektes ableiten. Allerdings sind diese zunächst nicht eindeutig, dadurch kann nicht sicher bestimmt werden, ob die Länge einer detektierten Kante zur Fahrzeugseite oder zu Front bzw. Heck des Fahrzeugs gehört. Sobald das Fahrzeug aber in Bewegung ist, kann aus der Trajektorie die Fahrtrichtung und damit die Länge und Breite des Objekts eindeutig zugeordnet werden.

### Objekt am Rand des Lidar-Sichtfeldes

Eine kritische Situation stellt ein Objekt dar, das sich am Rand des Lidar-Sichtfeldes befindet. In Abbildung 3-3 sind zwei Messungen eines am Fahrbahnrand stehenden Fahrzeuges abgebildet. Bei der ersten Messung befindet sich das Fahrzeug gerade noch vollständig im Sichtfeld der beiden Lidarsensoren (a), bei der anderen befindet sich das Objekt nur noch teilweise im Sichtfeld (b). Da die Sensoren nur die Seite des Fahrzeugs detektieren können, fehlen bei der zweiten Messung (b) Informationen über die Breite des Objekts und die berechnete Bounding Box ist kleiner als die Abmessungen des realen Fahrzeugs.

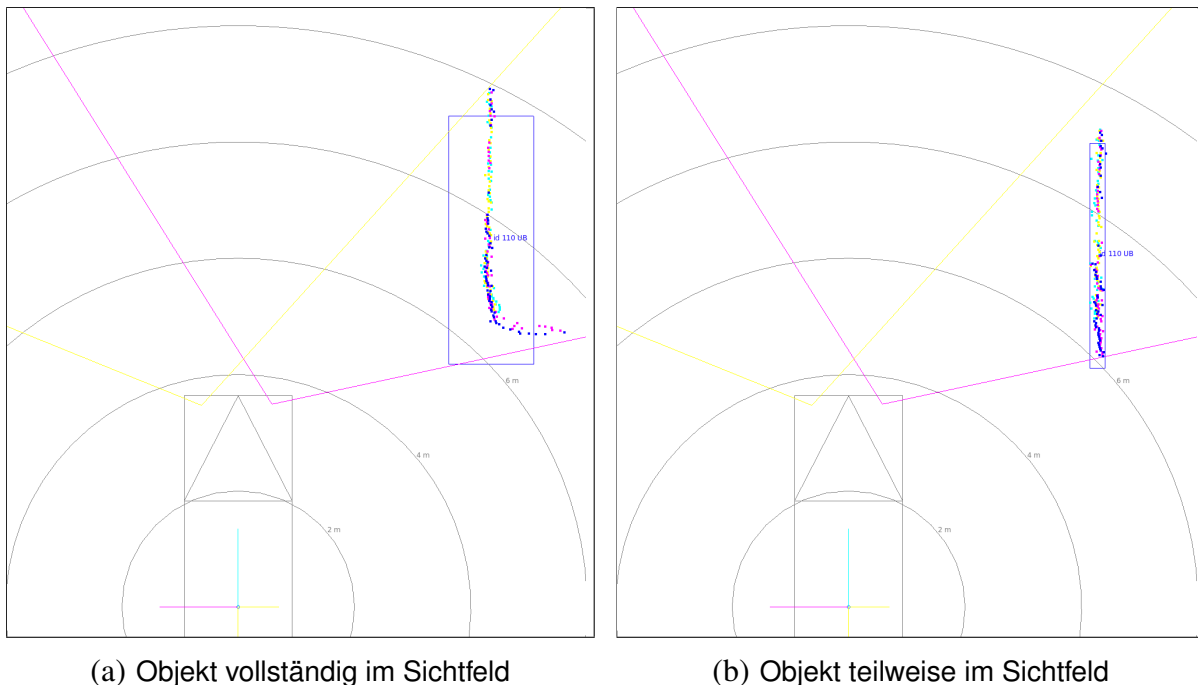


Abbildung 3-3: Zustandsunsicherheiten am Rand des Sensor-Sichtfeldes

## Wandernder Referenzpunkt

Die genaue Bestimmung des Referenzpunktes eines Objekts ist insbesondere für das Nachverfolgen von Fahrzeugen von hoher Bedeutung. Dieser kann u.a. zur Bestimmung der Geschwindigkeit verwendet werden.

Eine robuste Methode, den Referenzpunkt eines Objekts zu bestimmen, ist die Berechnung des geometrischen Schwerpunktes der segmentierten Punktwolke. Der Nachteil dieses Ansatzes ist in Abbildung 3-4 dargestellt. Je nach Blickwinkel auf das Objekt ergibt sich eine unterschiedliche Form der Punktwolke. Dies resultiert in einem kontinuierlich wandernden Referenzpunkt, der zu Fehlern bei der Positions- und Geschwindigkeitsermittlung führen kann.

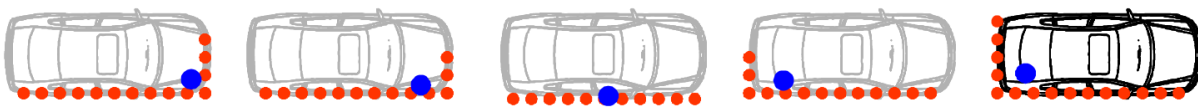


Abbildung 3-4: Wandernder Referenzpunkt.<sup>29</sup>

## Fehlerhafte Positionsbestimmung

In Abbildung 3-5 ist eine weitere Unsicherheit zu sehen. Die Länge und Breite des Objekts wird dabei korrekt bestimmt, die Bounding Box wird allerdings falsch positioniert. Die Ursache liegt in einer fehlerhaften Bestimmung der Objektposition. Diese wird als geometrischer Schwerpunkt der Punktwolke berechnet, dieser entspricht aber nie dem Mittelpunkt der Bounding Box.

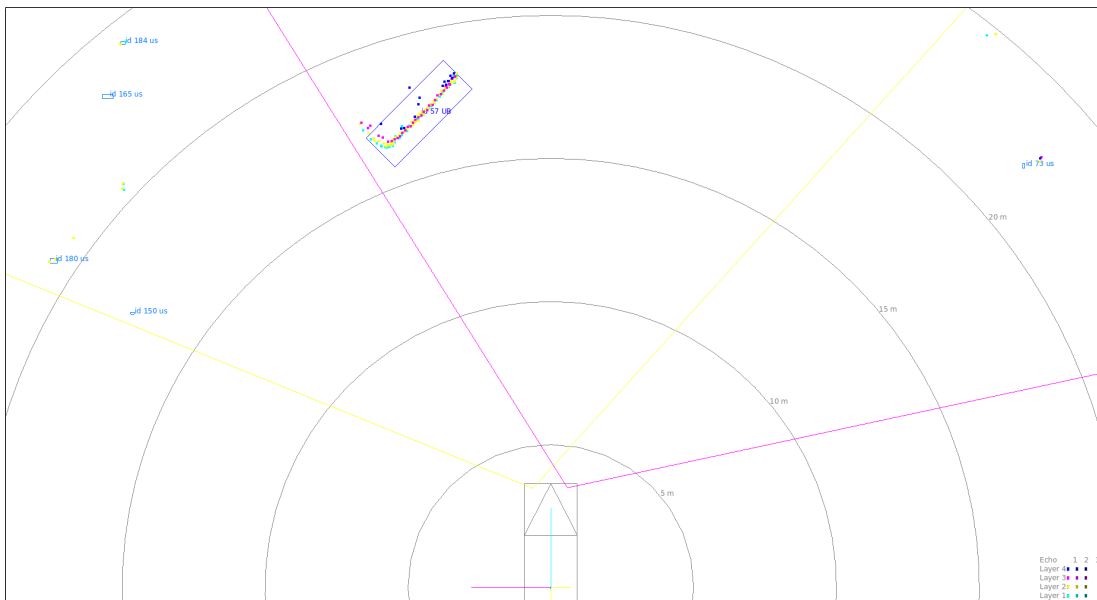


Abbildung 3-5: Fehlerhafte Positionsbestimmung

<sup>29</sup> Fürstenberg, K. C.: Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner (2009), S. 65.

## Verdeckung

Ein weiterer wichtiger Aspekt ist die Verdeckung. Wenn sich ein Objekt im Sichtfeld des Sensors befindet, können die Lidarstrahlen dahinter befindliche Objekte nur noch teilweise oder gar nicht mehr erreichen. Bei nur teilweiser Verdeckung verringern sich die wahrgenommenen Abmessungen der verdeckten Objekte. Abbildung 3-6 zeigt zwei Messungen, bei denen sich mehrere Objekte am rechten Fahrbahnrand befinden. Bei Messung (a) können alle Objekte nahezu vollständig erfasst werden, wobei in Messung (b) zwei Objekte verdeckt werden.

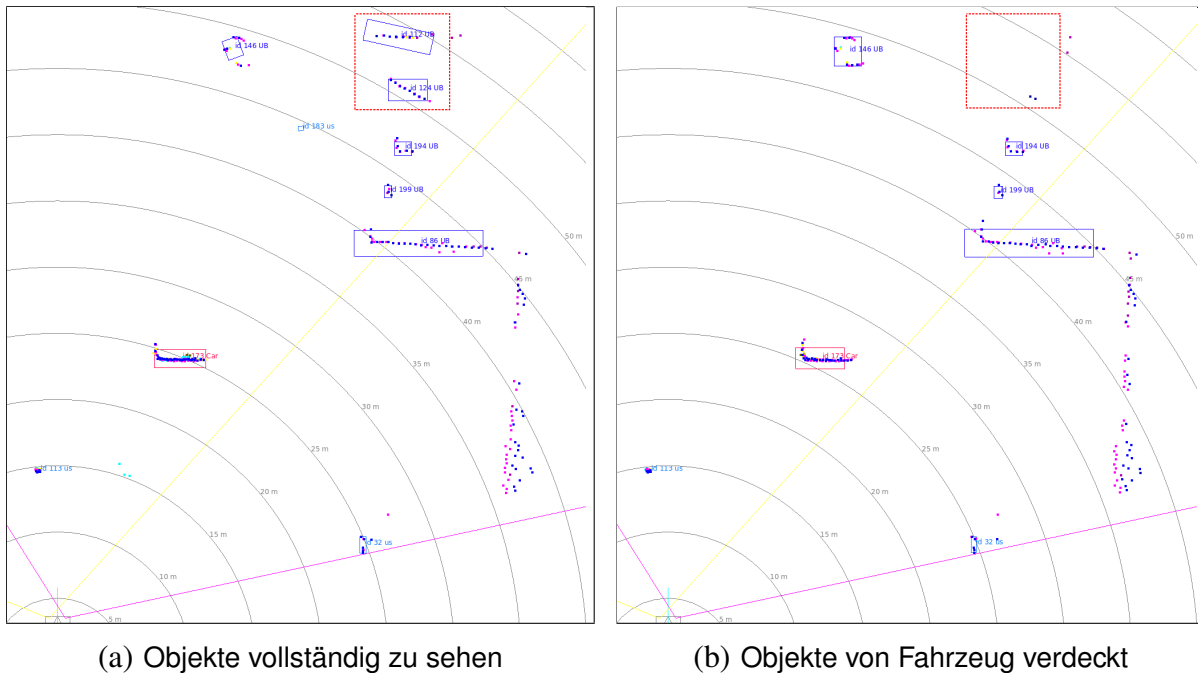


Abbildung 3-6: Verdeckung von Objekten

## Trennfähigkeit

Die Trennfähigkeit beschreibt die Eigenschaft, wie gut ein Sensor zwei sehr nahe beieinander befindliche Objekte als separate Objekte erkennen kann oder ob die Objekte als ein gemeinsames Objekt wahrgenommen werden. Die Trennfähigkeit ist abhängig von der Auflösung des Sensors und den Positionen der Objekte.

### 3.3 Klassenunsicherheiten

Die letzte Kategorie stellen Klassenunsicherheiten dar. Diese beschreiben die Unsicherheit bei der Zuordnung der semantischen Bedeutung zu den detektierten Objekten.

In Abbildung 3-7 werden beide real existierenden Fahrzeuge als Objekt erkannt, allerdings nur eines korrekt als Fahrzeug klassifiziert. Dem anderen Fahrzeug konnte keine Klasse zugeordnet werden und es wird daher als „Unknown Big“ gespeichert.

Abbildung 3-7: Beispiel für Klassenunsicherheiten

## 4 Implementierung des Modells

Im folgenden Kapitel wird die Vorgehensweise bei der Implementierung des Objektlisten-Modells erläutert. Zunächst wird die allgemeine Struktur des Modells vorgestellt. Die Basis bildet ein Modell, das eine ideale Objektliste ausgibt. Anschließend werden verschiedene Ansätze zur Modellierung des Ibeo Lux 2010 aufgezeigt, bewertet und die bestmögliche Methode bestimmt. Anschließend wird das Modell erweitert, sodass auch ein modernerer Lidarsensor mit verbesserten Algorithmen simuliert werden kann. Dabei wird insgesamt nur ein Modell erstellt, das parameterabhängig die verschiedenen Sensorvarianten realisiert.

### 4.1 Modellstruktur

In Abbildung 4-1 ist die Struktur des entwickelten Objektlisten-Sensormodells dargestellt. Aus der Simulationsumgebung, hier VTD, wird ein Datenstrom des Typs *osi::SensorData* ausgegeben und in das Modell eingelesen. Der Datenstrom enthält einerseits die GT-Informationen aus der Simulation und andererseits die Punktwolken der beiden simulierten Lidar-Sensoren in sphärischen Koordinaten. Ausgang des Modells ist wieder ein Datenstrom des Typs *osi::SensorData*, der neben der fusionierten Punktwolke in kartesischen Koordinaten die lidartypische Objektliste enthält.

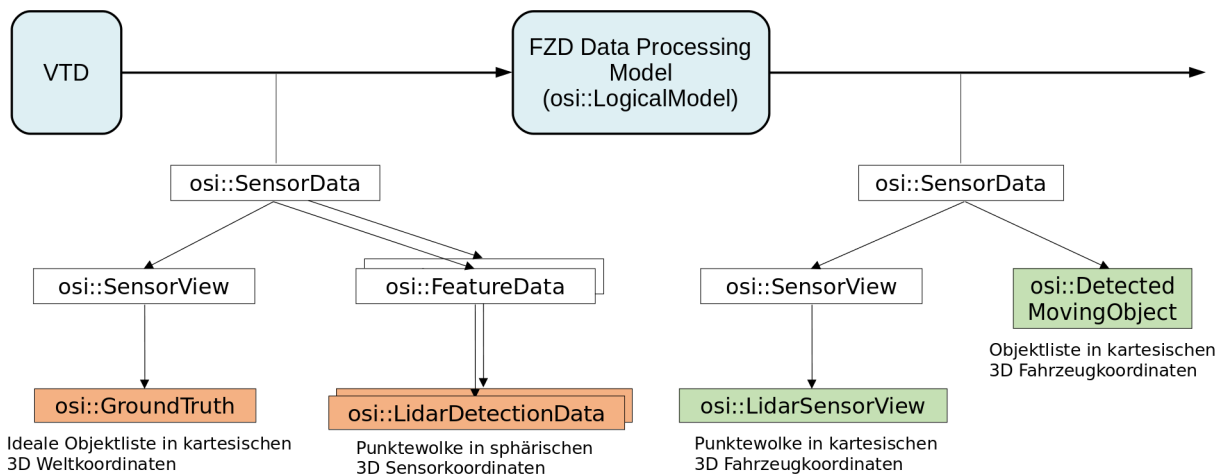


Abbildung 4-1: Struktur des Objektlisten-Sensormodells

### 4.2 Ideale Objektliste

Zunächst wird das Modell so ausgelegt, dass neben der fusionierten Punktwolke die ideale Objektliste ausgegeben wird. Dazu werden alle Eigenschaften der Objekte, die sich innerhalb des Sichtfeldes einer der beiden Sensoren befinden, aus der Ground Truth übernommen, vom Welt- ins Fahrzeug-Koordinatensystem transformiert und ohne Berücksichtigung weiterer Sensorunsicherheiten in den Ausgang geschrieben. Die ideale Objektliste kann später zum Vergleich mit der lidartypischen Objektliste des Ibeo-Modells und des erweiterten Modells genutzt werden.

---

### 4.3 Modellierung des Ibeo Lux 2010

Da bei FZD nur der Ibeo Lux 2010 als Lidarsensor zur Verfügung steht, der nicht nur Punktwolken, sondern auch Objektlisten liefert, wird ein Modell implementiert, das die wichtigsten Unsicherheiten dieses Sensors berücksichtigt. Der Vorgang zur Erstellung der Objektliste wird in die Schritte Segmentierung, Objektabmessungen, Klassifizierung und Tracking unterteilt. Anschließend wird auf die Implementierung der sensorspezifischen Unsicherheiten eingegangen. Im Folgenden werden verschiedene Ansätze für die Realisierung des jeweiligen Verarbeitungsschrittes unter Nutzung von Ground Truth und Punktwolken dargestellt und diskutiert.

#### 4.3.1 Modellierung der Segmentierung

Bei der Segmentierung besteht die Aufgabe darin, die fusionierte Punktwolke in Teilsegmente zu unterteilen und den jeweiligen Objekten zuzuordnen.

##### Modellbasierte Segmentierung

Eine Möglichkeit dazu bietet die modellbasierte Segmentierung. Dabei kann die Punktwolke auf primitive geometrische Formen untersucht werden. Da sich Objekte in den meisten Fällen als L-förmige Punktwolken darstellen, könnte nach dieser Form in der Punktwolke gesucht und jede einem Objekt zugeordnet werden (vgl. Abbildung 4-2).

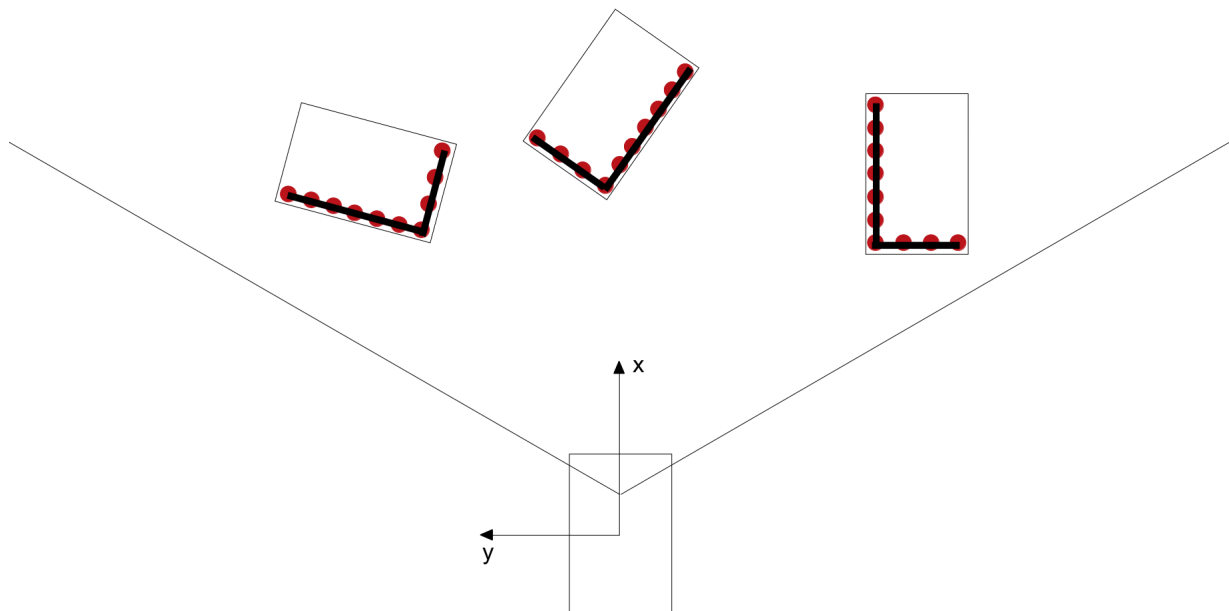


Abbildung 4-2: Modellbasierte Segmentierung



Diese Form der Segmentierung erfordert einen aufwendigen Algorithmus, der entsprechend eine große Rechenzeit benötigt. Da genau dies aber mit dem neuen Objektlisten-Modell vermieden werden soll, wird ein anderer Ansatz gesucht.

### Segmentierung mithilfe der Ground Truth

Bei einer Simulation bietet es sich an, die Ground Truth Informationen über existierende Objekte im Sichtfeld zu verwenden. Aus der bekannten Position und den geometrischen Abmessungen der Objekte kann der relevante Bereich direkt aus der Gesamtpunktwolke extrahiert werden. Dies kann über die Berechnung der minimalen und maximalen Ausdehnung in x- und y-Richtung realisiert werden (vgl. Abbildung 4-3 a). Alle Punkte, die sich innerhalb dieses Bereiches befinden, werden dem entsprechenden Objekt zugeordnet. Dabei muss die Orientierung  $\varphi$  des Zielobjektes berücksichtigt werden (siehe Abbildung 4-3 b).

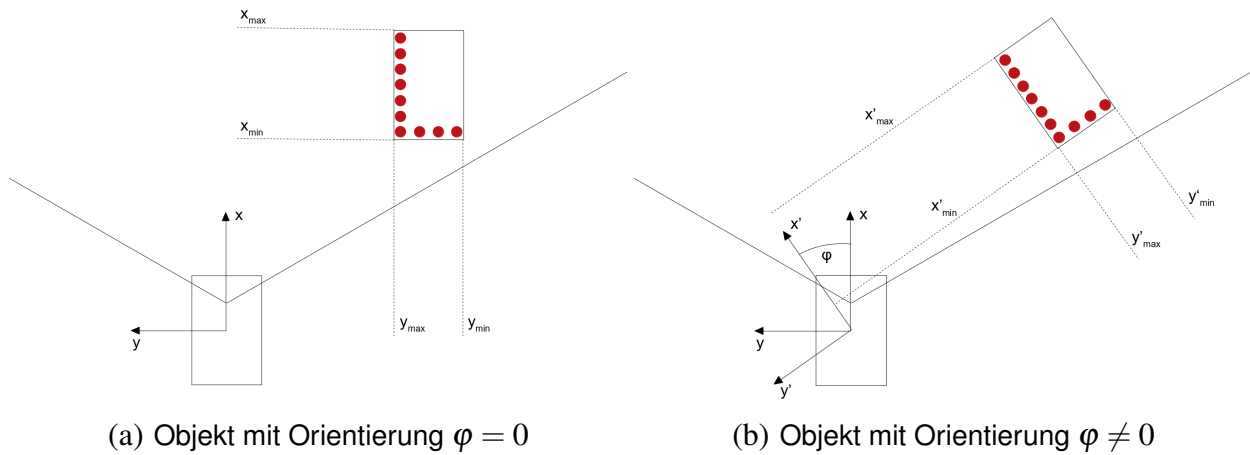


Abbildung 4-3: Segmentierung mittels Ground Truth Informationen

Da die hier verwendeten simulierten Punktwolken auf einem idealen Sensormodell basieren, sollte die Segmentierung mithilfe der GT Daten sehr robuste Ergebnisse liefern und kommt ohne einen rechenaufwendigen Algorithmus aus. Daher wird diese Methode im Objektlisten-Modell implementiert.

#### 4.3.2 Modellierung der Objektabmessungen

Eine weitere wichtige Aufgabe besteht darin, die Abmessungen eines Objektes zu bestimmen, um die Bounding Box des Objektes angeben zu können. Diese können entweder aus den Ground Truth Informationen oder der segmentierten Punktwolke bestimmt werden.

## Modellierung der Abmessungen mit Ground Truth

Bei der Bestimmung der Bounding Box ausschließlich mit Ground Truth Informationen werden zunächst die vier Eckpunkte des Objektes berechnet (blaue Markierungen in Abbildung 4-4). Anschließend werden die Eckpunkte (grün markiert) bestimmt, die sich im Sichtfeld des Sensors befinden und nicht verdeckt werden. Wenn sich ein Eckpunkt außerhalb des Sichtfeldes befindet, wird der für den Sensor äußerste sichtbare Punkt am Sichtfeldrand bestimmt. Außerdem muss die Verdeckung durch andere Objekte berücksichtigt werden.

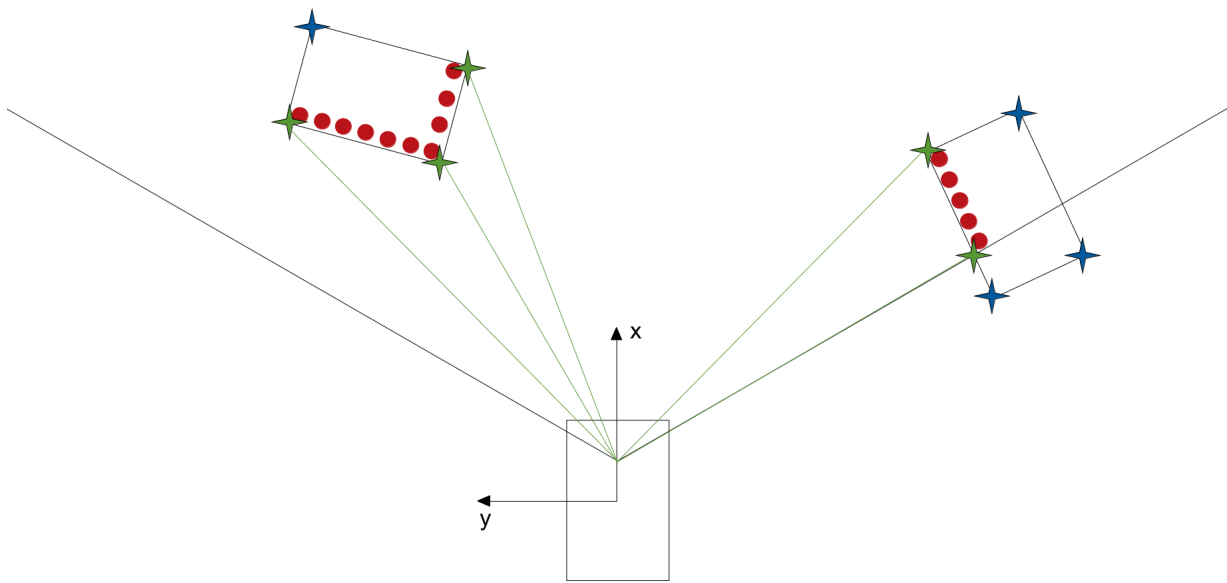


Abbildung 4-4: Bestimmung der Bounding Box mit Ground Truth

Da bei diesem Ansatz keine Punktwolken ausgewertet werden müssen, ist eine niedrige Rechenzeit zu erwarten, allerdings ist die Qualität der Ergebnisse abhängig von der Auflösung des verwendeten Sensors. Gerade beim Ibeo Lux 2010 mit nur vier Scanebenen würden die wahrgenommenen Abmessungen der Bounding Box in Realität und Simulation stark voneinander abweichen. Daher erscheint es sinnvoll, die zuvor segmentierte Punktwolke zu betrachten.

## Modellierung der Abmessungen mit segmentierter Punktwolke

Aus der segmentierten Punktwolke können sehr einfach die Abmessungen des Objektes bestimmt werden. Dazu muss nur die maximale Ausdehnung der Punktwolke in x- und y-Richtung bestimmt werden. Wie in Abbildung 4-5 dargestellt, muss dabei wieder die Orientierung  $\varphi$  des Objektes berücksichtigt werden.

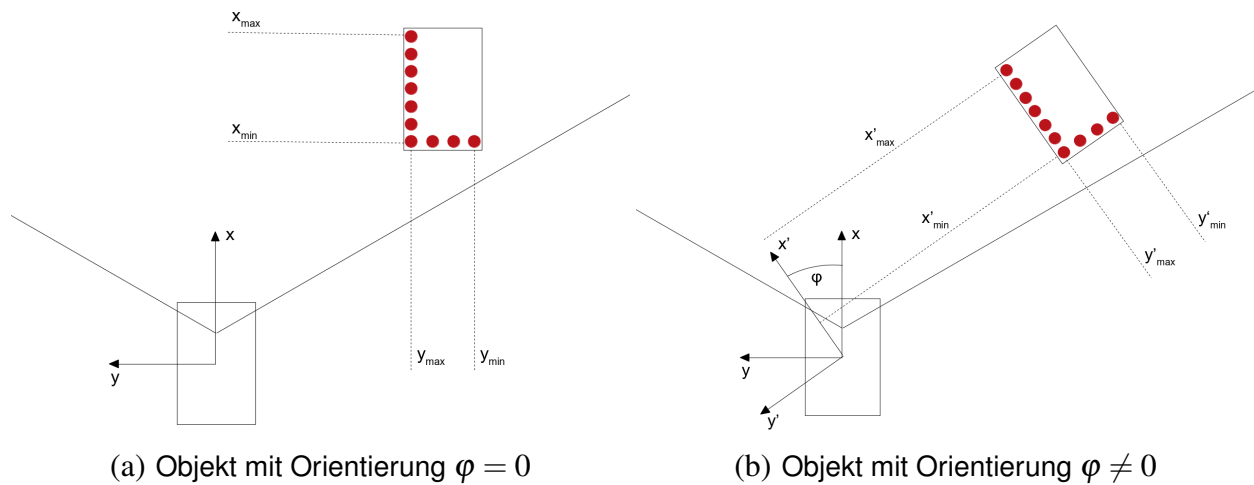


Abbildung 4-5: Bestimmung der Bounding Box mithilfe der segmentierten Punktwolke

Abhängig von der Qualität des Lidar-Sensormodells sind hierbei realitätsnahe Ergebnisse zu erwarten. Da bei dieser Methode ebenfalls keine hohe Rechenzeit zu erwarten ist, wird diese im Objektlisten-Modell implementiert. Ein weiterer Vorteil ist dabei, dass bei dieser Vorgehensweise automatisch die Verdeckung von Objekten sowie die Grenzen des Sensorsichtfelds berücksichtigt werden.

#### 4.3.3 Modellierung der Klassifizierung

Die Klassifizierung stellt eine komplexe Aufgabe dar. Diese könnte wie bisher mit den in Kapitel 2.2.4 erwähnten Algorithmen durchgeführt werden, was einer rechenlastigen Lösung entspricht, die möglichst vermieden werden soll. Andererseits zeigen Messungen des verwendeten Sensors, dass selbst dieser keine zufriedenstellenden Ergebnisse liefert. Der Großteil der Objekte kann nicht korrekt klassifiziert werden und wird als „Unknown Big“ oder „Unknown Small“ gekennzeichnet. Es können auch keine eindeutigen Bedingungen wie Position oder Orientierung identifiziert werden, bei denen die Klasse mehrheitlich korrekt bestimmt wird. Daher wird das Objektlisten-Modell zunächst so implementiert, dass allen detektierten Objekten die Klasse „Unknown Big“ zugeordnet wird, womit bereits eine hohe Übereinstimmung mit den realen Messungen des Ibeo Lux 2010 erreicht wird. Alternativ kann den Objekten auch die korrekte Klasse aus der idealen Objektliste übergeben werden.

Für zukünftige Arbeiten zur Erweiterung des Modells bietet sich die Implementierung einer merkmalsbasierten Klassifizierung an. Dabei könnten die wichtigsten Objekte wie Pkw, Lkw, Zweiräder und Fußgänger anhand ihrer Abmessungen klassifiziert werden. Neben diesen dynamischen Objekten ist es ebenfalls wichtig, statische Objekte wie Gebäude zu erkennen. Dazu muss die absolute Geschwindigkeit der Objekte bestimmt werden. Diese Art der Klassifizierung benötigt im Gegensatz zur End-to-end-Methode weitaus weniger Rechenzeit und wäre daher ideal für ein Objektlisten-Modell.

#### 4.3.4 Tracking

Entscheidend für den Algorithmus zur zeitlichen Verfolgung der Objekte sind die Auslegungen, ab welchem Zeitpunkt ein Objekt initialisiert und unter welchen Bedingungen es wieder aus dem Track-Management gelöscht wird. Die bisherigen Tracking-Algorithmen am Fachgebiet FZD<sup>30,31</sup> sind so ausgelegt, dass ein Objekt nach fünf aufeinander folgenden Detektionen initialisiert wird und wieder gelöscht wird, wenn es in zehn aufeinander folgenden Messungen nicht detektiert wurde. Da die Segmentierung im Objektlisten-Modell auf den GT-Daten basiert, werden selbst einzelne Punkte dem korrekten Objekt zugeordnet, was nicht dem realen Sensorverhalten entspricht. Daher müssen neue Bedingungen für eine Objekterkennung bzw. Start und Ende des Tracking bestimmt werden.

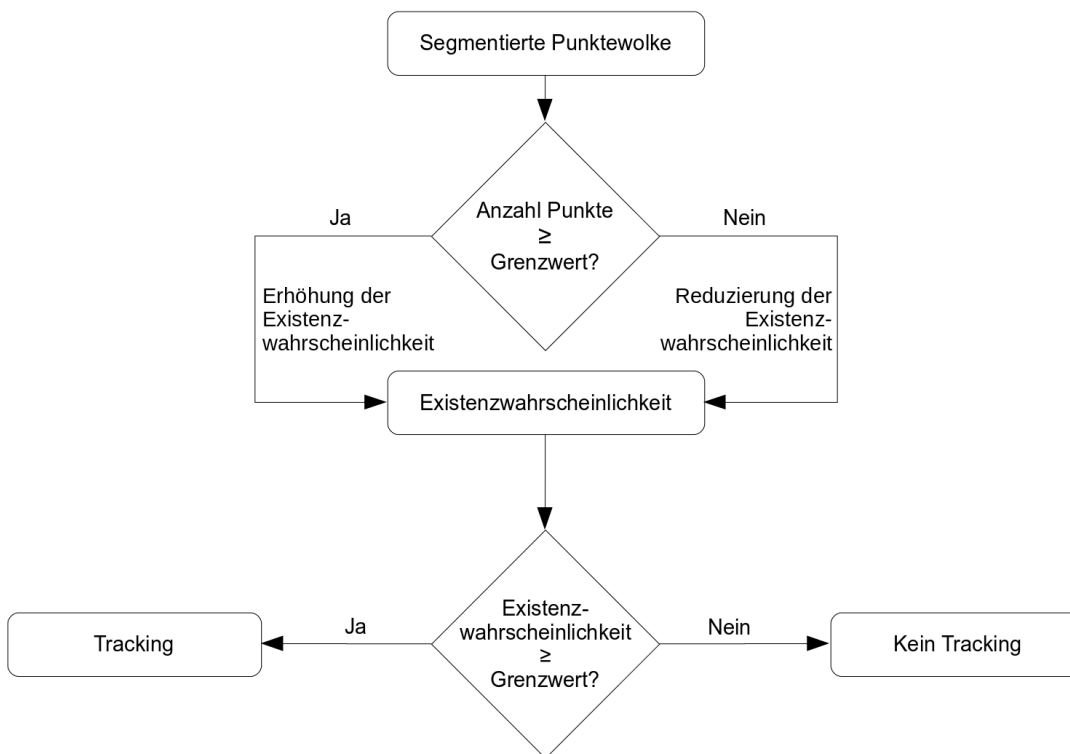


Abbildung 4-6: Methodik des implementierten Tracking-Algorithmus

Eine Möglichkeit ist die Initialisierung und Löschung von Objekten in Abhängigkeit der Größe der segmentierten Punktwolke, welche für dieses Modell genutzt wird. Abbildung 4-6 zeigt die Methodik des implementierten Tracking-Algorithmus. Dabei wird jedem Objekt eine Existenzwahrscheinlichkeit zugeordnet, die sich ab einer vorgegebenen Minimalgröße der segmentierten Punktwolke um 10 % erhöht. Ist die Punktwolke kleiner, wird die Existenzwahrscheinlichkeit reduziert. Da dynamische Objekte leichter zu detektieren sind als statische, wird eine Fallunterscheidung implementiert. Wenn die Absolutgeschwindigkeit eines Objektes (aus Ground Truth Daten bekannt) über einem festgelegten Wert liegt, wird die Mindestgröße der Punktwolke für die Erhöhung der Existen-

<sup>30</sup> Knerr, J.: Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018)

<sup>31</sup> Zhang, Y.: Implementation of an enhanced Lidar Tracking algorithm for automated driving (2018)

---

zwahrscheinlichkeit reduziert. Überschreitet die Existenzwahrscheinlichkeit einen Grenzwert, wird das Objekt schließlich initialisiert und in der Objektliste ausgegeben. Messungen mit dem Ibeo Lux 2010 haben gezeigt, dass ein bereits detektiertes Objekt so lange verfolgt wird, bis kein einziger Punkt mehr dem Objekt zugeordnet werden kann. Daher wird im implementierten Modell die Mindestgröße der Punktwolke zur Erhöhung der Existenzwahrscheinlichkeit auf eins gesetzt, sobald die Existenzwahrscheinlichkeit über dem Grenzwert liegt und das Objekt im Track-Management initialisiert wird.

#### **4.3.5 Sensorspezifische Unsicherheiten**

Um mit der Simulation realitätsnahe Ergebnisse zu erzielen, werden im Folgenden Ansätze vorgestellt, um die in Kapitel 3 aufgezeigten sensorspezifischen Unsicherheiten des Ibeo Lux 2010 im Modell zu implementieren.

##### **Positionsbestimmung**

Wie in Kapitel 3.2 gezeigt, bestimmt der Ibeo Lux 2010 die Position  $P(t)$  des Objektes als Schwerpunkt der segmentierten Punktwolke. Um die Ausgabe des Sensors möglichst realitätsnah nachzubilden, wird die Position im Objektlistenmodell ebenfalls als geometrischer Schwerpunkt der segmentierten Punktwolke berechnet. Dies resultiert in einer Position, die abhängig vom Blickwinkel auf das Objekt ist. In Kapitel 4.4.1 wird ein Ansatz vorgestellt, mit dem die Objektposition genauer bestimmt werden kann.

##### **Geschwindigkeitsbestimmung**

Die Geschwindigkeit wird durch Ableitung der Position in X-, Y- und Z-Richtung mit Hilfe des Rückwärtsdifferenzenquotienten 1. Ordnung berechnet:

$$v(t) = \frac{P(t) - P(t - \Delta t)}{\Delta t} \quad (4-1)$$

Durch den wandernden Referenzpunkt sind bei dieser Vorgehensweise starke Abweichungen von der idealen Geschwindigkeit zu erwarten. Um die Schwankungen zu reduzieren, wird der Mittelwert der letzten zehn Messungen gebildet. Vorteilhaft wäre die Ableitung eines ortsfesten Punktes, beispielsweise eines Eckpunktes des Objektes, welche in Kapitel 4.4.2 vorgestellt wird.

##### **Zuordnung von Länge und Breite bei stehenden Objekten**

Wie in Kapitel 4.3.2 dargestellt, können die Dimensionen eines Objektes unter Berücksichtigung dessen Orientierung einfach bestimmt werden. Ist die Orientierung des Objektes allerdings nicht bekannt, gestaltet sich die Bestimmung der Länge und Breite schwierig. Aus der Form der Punktwolke lassen sich zwar erste Informationen über die Orientierung und damit die Abmessungen des Objektes ableiten, allerdings können diese nicht sicher der Länge und Breite des Objektes zugeordnet

werden (vgl. Abbildung 4-7). Sobald sich das Objekt aber bewegt, kann aus dessen Trajektorie die Bewegungsrichtung bestimmt werden, wodurch die Länge und Breite eindeutig zugeordnet werden können.

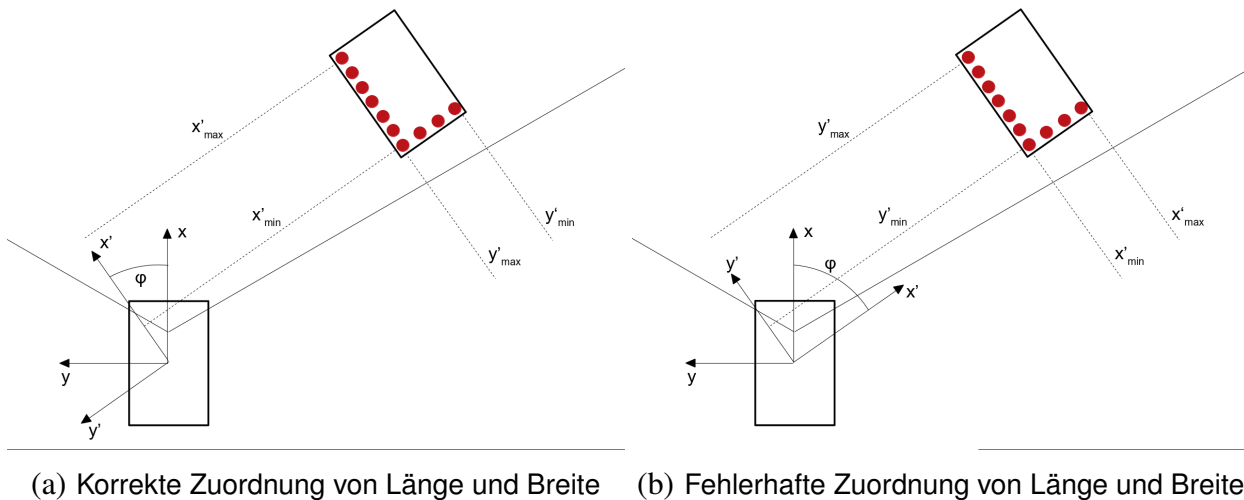


Abbildung 4-7: Zuordnung von Objektlänge und -breite bei stehenden Objekten

Die Unsicherheit wird folgendermaßen implementiert: Bei stehenden Objekten wird zunächst davon ausgegangen, dass sie ähnlich zum Ego-Fahrzeug ausgerichtet sind. Das heißt, bei Orientierungen von  $-90^\circ \leq \varphi \leq 90^\circ$  werden Länge und Breite korrekt zugeordnet, ansonsten werden die Größen vertauscht. Ist das Objekt dagegen in Bewegung, werden die Abmessungen immer korrekt zugeordnet. Wird die Orientierung eines Objektes einmal eindeutig bestimmt, werden die Abmessungen so lange korrekt zugeordnet, bis das Objekt aus dem Track-Management gelöscht wird.

## Trennfähigkeit

Die Trennfähigkeit ist abhängig von der Auflösung des verwendeten Sensors. Eine Möglichkeit zur Berücksichtigung der Trennfähigkeit ist die Bestimmung einer Zone um das Zielobjekt, in der die Trennfähigkeit gewährleistet ist. Die Größe dieser Zone wäre abhängig von der Sensorauflösung und der Entfernung des Objektes. Eine Alternative besteht darin, den Erfassungsbereich des Sensors in feste Bereiche zu untergliedern, in denen ein Objekt unter bestimmten Bedingungen von anderen Objekten getrennt werden kann, oder ob sie als ein gemeinsames Objekt wahrgenommen werden. Im Modell wird die Trennfähigkeit noch nicht implementiert, die Trennfähigkeit des Sensors wird als ideal angenommen.

## 4.4 Erweiterung des Ibeo-Modells

Um neben dem Ibeo Lux 2010 auch weitere Sensoren mit besseren Algorithmen simulieren zu können, werden im Folgenden Ansätze zur genaueren Bestimmung der Objektposition, -geschwindigkeit

und -abmessungen vorgestellt, die durch eine einfache Änderung der Modellparameter angewandt werden können.

#### 4.4.1 Bestimmung der Position

Zur genaueren Bestimmung der Position eines Objektes wird zunächst ein Eckpunkt der Bounding Box als Referenzpunkt bestimmt (vgl. Abbildung 4-8). Dessen Position kann abhängig von der Orientierung des Objektes aus der maximalen Ausdehnung der segmentierten Punktwolke berechnet werden. Ausgehend von diesem Eckpunkt wird die Hälfte der zuvor bestimmten Länge und Breite des Objektes auf den Referenzpunkt addiert. Je genauer die Abmessungen des Objektes berechnet werden können, desto näher liegt die bestimmte Position an der idealen Position.

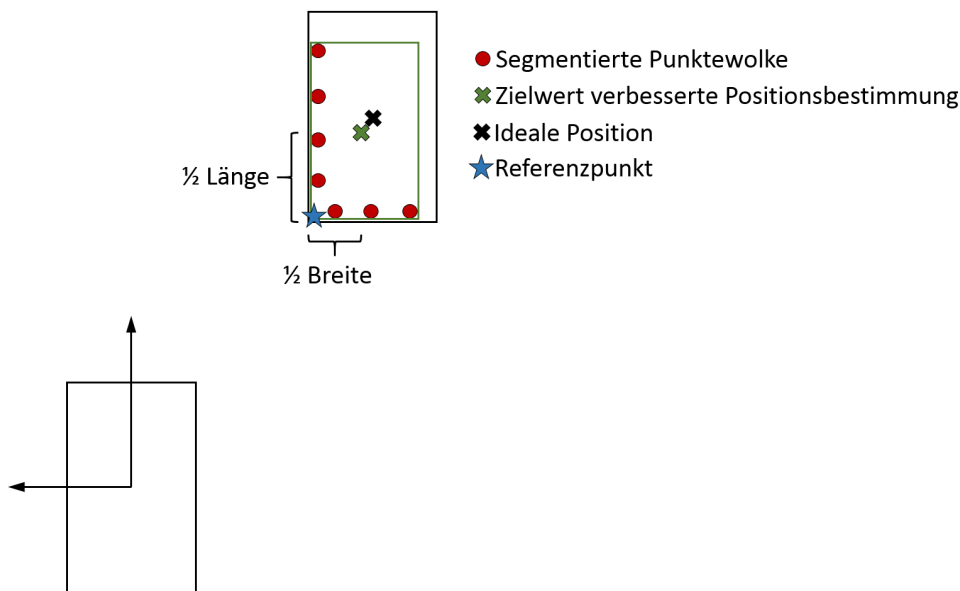


Abbildung 4-8: Methodik der verbesserten Positionsbestimmung

Falls das Objekt teilweise verdeckt wird, kann es bei der Modellierung zu Fehlern bei der Bestimmung des Referenzpunktes kommen, die sich auf die Berechnung der Objektposition auswirken. Da sich in diesem Fall aber ebenfalls die ermittelten Objektanmessungen reduzieren, hält sich der Fehler in Grenzen.

#### 4.4.2 Bestimmung der Geschwindigkeit

Durch die Ableitung eines wandernden Referenzpunktes wird die Geschwindigkeit verfälscht. Daher soll im erweiterten Ibeo-Modell die Position eines ortsfesten Punktes am Objekt abgeleitet werden. Dazu bietet sich der Referenzpunkt an, der auch zur verbesserten Positionsbestimmung verwendet wird (vgl. Abbildung 4-8). Dadurch sollten die Geschwindigkeiten geringere Schwankungen aufweisen. Zusätzlich wird wie beim Ibeo-Modell der Mittelwert über die vergangenen zehn Messungen

---

gebildet. Allerdings kann es zu Ausreißern kommen, wenn der Referenzpunkt aufgrund einer veränderten Orientierung des Objektes auf eine andere Ecke springt.

#### 4.4.3 Bestimmung der Abmessungen

Die Genauigkeit der Objektabmessungen aus den Punktwolken ist abhängig vom aktuellen Blickwinkel der Sensoren auf das Objekt, der Entfernung zum Objekt und dem Verdeckungsgrad. Daher könnte das Tracking Objektabmessungen, die zuvor bereits genau bestimmt wurden, für einen bestimmten Zeitraum speichern, um sie wieder abrufen zu können, wenn die Abmessungen zu einem späteren Zeitpunkt schlechter bestimmt werden können. Dazu wird eine Funktion implementiert, die zu jeder Messung einen Qualitätswert für die Länge und Breite eines Objektes berechnet. Im Tracking werden anschließend die aktuellen Qualitätswerte mit denen des letzten Zeitpunktes verglichen: Ist der Qualitätswert der aktuellen Messung höher, wird die berechnete Länge oder Breite zusammen mit dem Qualitätswert gespeichert, wenn nicht werden die Werte des letzten Zeitschrittes übernommen.

Die Berechnung des Qualitätswertes wird in drei verschiedene Teilfaktoren unterteilt: Den Blickwinkel auf das Objekt  $Q_{L,b}$  und  $Q_{B,b}$ , die Entfernung zum Objekt  $k_d$  und die Verdeckung  $k_V$ .

$$\begin{aligned} Q_L &= Q_{L,b} \cdot k_d \cdot k_V \\ Q_B &= Q_{B,b} \cdot k_d \cdot k_V \end{aligned} \tag{4-2}$$

##### Blickwinkel auf das Objekt

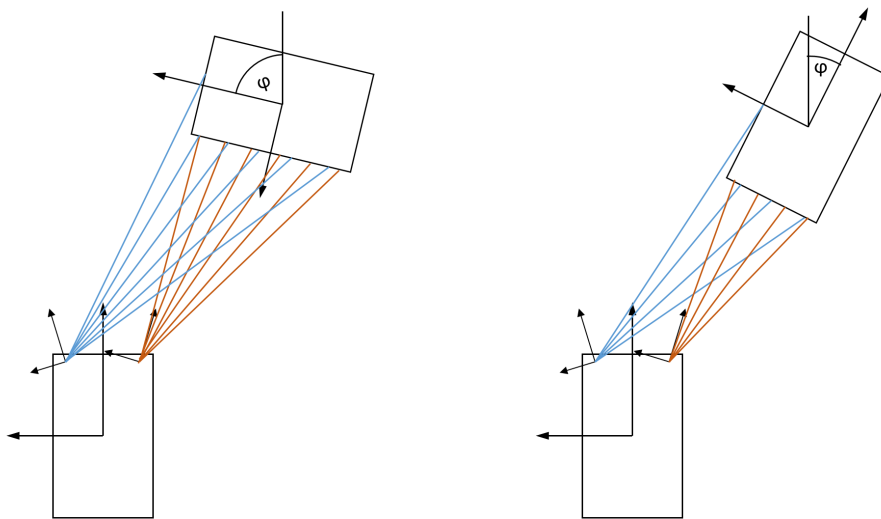
Die Basis bildet die Ansicht auf das Objekt. Der Blickwinkel von einem Sensor zum Objekt wird als Differenz zwischen der Orientierung des Objekts  $\varphi$  und dem Azimutwinkel  $\alpha$  vom Sensor zum Objekt definiert. Ist ein Fahrzeug längs zum Blickwinkel des Sensors ausgerichtet, erreichen viele Strahlen das Heck oder die Front und die wahrgenommene Breite ist nah an der Realität, gleichzeitig ist die Bestimmung der Länge bei dieser Ansicht sehr ungenau (vgl. Abbildung 4-9 b). Ist das Fahrzeug quer zum Blickwinkel des Sensors orientiert, kann die Länge gut bestimmt werden, die Breite dagegen nicht.

Der Basis-Qualitätswerte für die Länge und Breite  $Q_{L,b}$  und  $Q_{B,b}$  werden wie folgt berechnet:

$$\begin{aligned} Q_{L,b} &= \sin(\varphi - \alpha) \\ Q_{B,b} &= \cos(\varphi - \alpha) \end{aligned} \quad \text{mit } \alpha = \tan^{-1} \left( \frac{y}{x} \right) \tag{4-3}$$

Befindet sich das Objekt im Sichtfeld beider Sensoren, wird der Mittelwert der Qualitätswerte für beide Sensoren gebildet.





(a) Vorteilhafte Ansicht für die Bestimmung der Länge

(b) Vorteilhafte Ansicht für die Bestimmung der Breite

Abbildung 4-9: Bestimmung eines Qualitätswerts abhängig vom Blickwinkel

### Entfernung des Objektes

Die Genauigkeit der berechneten Objektanmessungen hängt zudem von der radialen Entfernung des Objektes ab. Je weiter ein Objekt entfernt ist, desto größer sind die horizontalen und vertikalen Abstände zwischen zwei Laserstrahlen. Somit treffen weniger Strahlen das Objekt und die Genauigkeit der Objektanmessung nimmt ab. Daher wird der Korrekturfaktor  $k_d$  eingeführt, um Abmessungen zu priorisieren, die bei einer geringen Distanz zum Objekt berechnet werden. Der Faktor (siehe Abbildung 4-10) wird folgendermaßen berechnet:

$$k_d = \begin{cases} k_d = 1 - (0,5 \cdot d/100) & , \text{ wenn } d < 100 \text{ m} \\ 0,5 & \text{sonst} \end{cases} \quad (4-4)$$

mit

$$d = \sqrt{x^2 + y^2}. \quad (4-5)$$

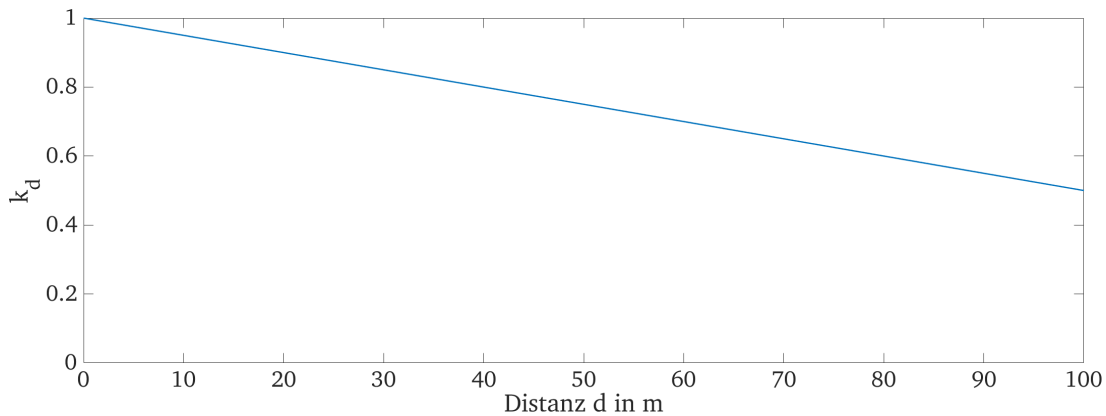


Abbildung 4-10: Korrekturfaktor für die Objektentfernung

### Verdeckung eines Objektes

Wichtig für die Bildung des Qualitätswertes ist außerdem der Grad der Verdeckung eines Objektes. Denn ist der Blickwinkel auf ein Objekt sehr gut zur Bestimmung der Objektlänge oder -breite, ist der Basis-Qualitätswert hoch, obwohl das Objekt gar nicht oder nur teilweise vom Sensor erfasst werden kann. Daher muss die Verdeckung berücksichtigt werden, damit die Abmessungen nicht dauerhaft gespeichert werden.

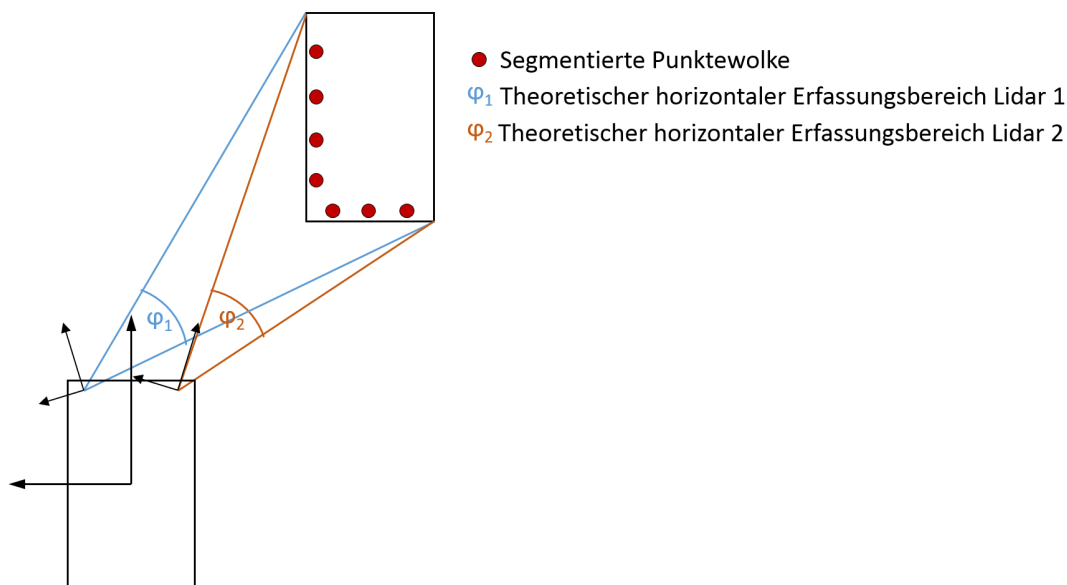


Abbildung 4-11: Bestimmung des Verdeckungsgrades eines Objektes

In Abbildung 4-11 ist der Ansatz zur Berechnung des Korrekturfaktors abgebildet. Aus der Ground Truth wird für jeden Sensor der horizontale Azimut-Bereich berechnet, den das Objekt im aktuellen Zustand einnimmt. Teilt man diesen Winkel durch den Winkel zwischen zwei Strahlen des Laserscanners, kann man bestimmen, wie viele Strahlen des Sensors maximal das Zielobjekt treffen können. Die gleiche Methode wird für den Elevationswinkel angewandt. Werden die maximal erreichbaren

Punkte in horizontaler und vertikaler Richtung miteinander multipliziert, erhält man einen Näherungswert für die maximal erreichbare Größe der segmentierten Punktwolke pro Sensor. Teilt man nun die tatsächliche Größe der segmentierten Punktwolke  $N_{seg}$  durch die zuvor berechnete maximal mögliche Punkteanzahl  $N_{seg,max}$ , erhält man ein Maß für die Verdeckung eines Objektes:

$$k_V = \frac{N_{seg}}{N_{seg,max}} \quad (4-6)$$

mit

$$N_{seg,max} = N_{h,1} \cdot N_{v,1} + N_{h,2} \cdot N_{v,2} = \frac{\varphi_{h,1}}{\delta_h} \cdot \frac{\varphi_{v,1}}{\delta_v} + \frac{\varphi_{h,2}}{\delta_h} \cdot \frac{\varphi_{v,2}}{\delta_v} \quad (4-7)$$

#### 4.5 Übersicht über die verschiedenen Modellvarianten

In Tabelle 4-1 gibt eine Übersicht darüber, wie die verschiedenen Modellvarianten die Eigenschaften eines Objektes bestimmen.

Tabelle 4-1: Ausgabe der verschiedenen Modellvarianten

<b>Variable</b>	<b>Ideale Objektliste</b>	<b>Ibeo-Modell</b>	<b>Erweitertes Modell</b>
Klasse	Ideal	Unkown	Unkown
Position	Ideal (Mittelpunkt der Bounding Box)	Geom. Schwerpunkt der segmentierten Punktwolke	Bestimmung aus Referenzpunkt und Objektabmessungen
Geschwindigkeit	Ideal	Ableitung der Position	Ableitung des Referenzpunktes
Beschleunigung	Ideal	Ideal	Ideal
Orientierung	Ideal	Ideal	Ideal
Abmessungen	Ideal	Aus segmentierter Punktwolke	Aus segmentierter Punktwolke und Qualitätswert

---

## 5 Evaluation des Modells

---

Zur stichprobenartigen Validierung des entwickelten Objektlisten-Modells werden reale Fahrversuche durchgeführt, die im Folgenden beschrieben werden. Anschließend werden die Versuche in VTD nachsimuliert und die realen und synthetischen Sensordaten mittels geeigneter Metriken für Objektlisten verglichen.

### 5.1 Durchführung von Fahrversuchen

Zum Aufzeichnen realer Sensordaten werden auf dem August-Euler-Flugplatz in Griesheim, dem Testgelände der TU Darmstadt, Fahrversuche durchgeführt. Dazu wird zunächst der Lidarsensor kalibriert, anschließend werden zwei Szenarien nachgestellt. Während den Versuchen werden neben den Lidardaten die GPS-Position sowie die Ausrichtung der bewegten Fahrzeuge gemessen, sodass die Szenarien anschließend in VTD nachsimuliert werden können.

#### 5.1.1 Kalibrierung

Da bei früheren Versuchen Abweichungen der Ausrichtung der Sensoren von der Norm festgestellt wurden, wird zunächst die horizontale Orientierung der beiden Sensoren experimentell bestimmt und kalibriert. Dazu wird das Versuchsfahrzeug mittig entlang einer Linie auf einer freien Fläche positioniert und ausgerichtet (vgl. Abbildung 5-1). Auf der Linie wird im Abstand von 50 m ein Reflektor platziert. Da der Reflektor sehr schmal ist, wird er in den Punktwolken der beiden Sensoren jeweils als einzelner Punkt angezeigt.

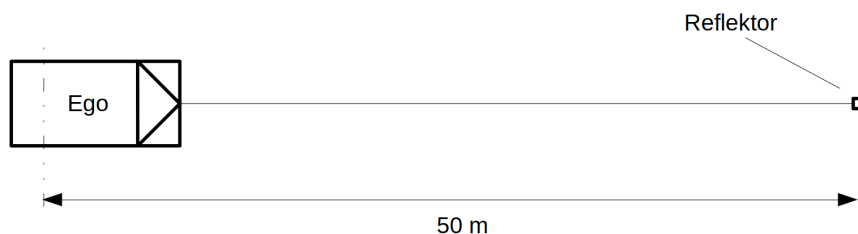


Abbildung 5-1: Kalibrierung der Lidarsensoren

Anschließend wird der Winkel, mit dem die Software des Sensors die Punktwolke in das kartesische Koordinatensystem des Fahrzeugs transformiert, schrittweise angepasst, bis der Reflektor in der Punktwolke genau mittig angezeigt wird. Danach wird der Vorgang mit dem zweiten Sensor wiederholt. Die kalibrierten Winkel (siehe Tabelle 5-1) werden im Setup der Simulation übernommen.

Tabelle 5-1: Kalibrierte Gierwinkel der Lidarsensoren

	Ursprüngliche Gierwinkel	Kalibrierte Gierwinkel
<b>Lidar 1</b>	17,906°	19,3°
<b>Lidar 2</b>	-17,906°	-16,8°

### 5.1.2 Versuch 1

In beiden Versuchen soll eine alltägliche Situation an einer Kreuzung dargestellt werden. Der Aufbau des ersten Szenarios ist in Abbildung 5-2 dargestellt. Der VW Crafter steht an einer Kreuzung, das Ego-Fahrzeug 12,60 m dahinter, beide Fahrzeuge werden während des gesamten Versuchs nicht bewegt. Der BMW i3 steht zunächst in 30 m Abstand zur Kreuzung auf der kreuzenden Straße, beschleunigt auf 10 km/h, biegt links ab und passiert schließlich Crafter und Ego-Fahrzeug. Während des Abbiegevorgangs wird der i3 kurzzeitig durch den Crafter verdeckt.

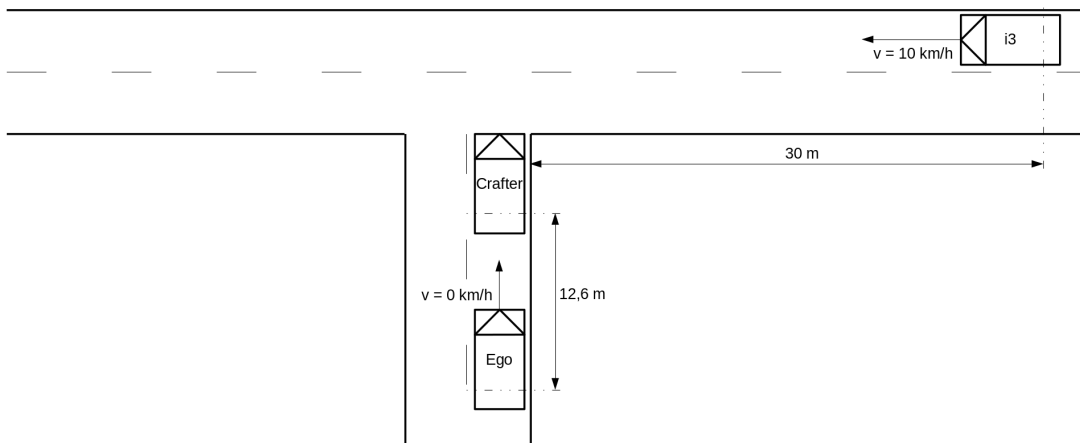


Abbildung 5-2: Aufbau des ersten Szenarios

Mit dem Szenario können die wichtigsten Merkmale in einer Objektliste bewertet werden: Die Position, Orientierung, Relativgeschwindigkeit und geometrischen Abmessungen. Das Szenario enthält stehende sowie bewegte Objekte unterschiedlicher Größe in besonders relevanten Positionen (vorausfahrendes, querendes und entgegenkommendes Fahrzeug). Zudem wird mit der kurzzeitigen Verdeckung des i3 die Wahrnehmung erschwert, ein Phänomen, das in der Realität häufig auftritt.

### 5.1.3 Versuch 2

Der zweite Versuch ist ähnlich zum ersten Szenario aufgebaut. Crafter und i3 befinden sich in derselben Startposition wie in Versuch 1, das Ego-Fahrzeug steht 45 m hinter dem Crafter. Anschließend beschleunigen i3 und Ego-Fahrzeug gleichzeitig auf 10 km/h. Das Ego-Fahrzeug bleibt kurz hinter dem Crafter stehen, während der i3 wie im ersten Versuch links abbiegt und beide Fahrzeuge passiert.



Abbildung 5-3: Aufbau des zweiten Szenarios

## 5.2 Simulation der Fahrversuche

Zur Simulation der realen Fahrversuche werden die GPS-Daten der bewegten Fahrzeuge in Trajektorien umgewandelt, die den jeweiligen Fahrzeugen in der Simulation zugeordnet werden können. Position und Ausrichtung des statischen Crafters werden mit Maßband und Kompass bestimmt und in die Simulationsumgebung übertragen. Da für den VW Crafter in VTD kein Fahrzeugmodell verfügbar ist, wird stattdessen auf ein Modell eines Mercedes-Benz Sprinters zurückgegriffen (vgl. Tabelle

5-2), der ähnliche geometrische Abmessungen besitzt. Das Ego-Fahrzeug (Honda Accord) befindet sich nicht im Sichtfeld der Sensoren und benötigt daher kein äquivalentes Modell. Allerdings ist zu beachten, dass die Positionen und Orientierungen der Sensoren mit denen des realen Ego-Fahrzeugs übereinstimmen müssen.

Tabelle 5-2: Verwendete Versuchsfahrzeuge und Äquivalente in VTD

Fahrzeug im realen Fahrversuch	Fahrzeugäquivalent in VTD
BMW i3	BMW i3
VW Crafter	Mercedes-Benz Sprinter



Abbildung 5-4: Versuch 1 in VTD

## 5.3 Metriken

Die realen und simulierten Objektlisten sollen anschließend anhand geeigneter Metriken verglichen werden. Huch<sup>32</sup> stellt dazu in seiner Masterthesis eine Reihe von Metriken vor. Für die Bewertung des Objektlisten-Sensormodells werden Root Mean Square Error, Intersection over Union und die OSPA-MT-Metrik ausgewählt, die im Folgenden erläutert werden.

### 5.3.1 Root Mean Square Error

Eine Möglichkeit zur Bewertung des Modells stellt die Berechnung des Root Mean Squared Errors (RMSE) zwischen zwei Mengen mit Trajektorien dar.<sup>33</sup> Dabei kann der RMSE sowohl für die Positionen, als auch für die Geschwindigkeiten und Abmessungen berechnet werden. Wenn mehrere Trajektorien der einen Menge mit einer Trajektorie der anderen Menge korrespondieren, wird eine Kombination der Trajektorien verwendet. Dies erfordert eine manuelle Bearbeitung der Trajektorien vor der Auswertung.

<sup>32</sup> Huch, S.: Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation (2018)

<sup>33</sup> Morton, P.; Douillard, B.; Underwood, J.: An evaluation of dynamic object tracking with 3D LIDAR (2011)



### 5.3.2 OSPA-MT

Die OSPA-MT-Metrik<sup>34</sup> bewertet die Distanz zwischen zwei Mengen mit Trajektorien. Die OSPA-MT-Distanz zwischen den Teilmengen  $\omega$  und  $\omega'$  mit der Ordnung  $p$  und dem Cut-Off-Parameter  $C$ , wobei  $1 \leq p < \infty$  und  $0 < \Delta < C$ , ist definiert als

$$d_{C,p}^{\Delta}(\omega, \omega') = \min\{d_{C,p}^{\Delta}(\overrightarrow{\omega}, \overrightarrow{\omega'}), d_{C,p}^{\Delta}(\overrightarrow{\omega'}, \overrightarrow{\omega})\} \quad (5-1)$$

mit der Quasi-OSPA-MT-Metrik

$$d_{C,p}^{\Delta}(\overrightarrow{\omega}, \overrightarrow{\omega'}) = \begin{cases} \min_{\lambda \in M(L^{\omega'}, L_0^{\omega})} \tilde{d}_{C,p}^{\Delta, \lambda}(\overrightarrow{\omega}, \overrightarrow{\omega'}) & , \text{ wenn } \omega, \omega' \neq \emptyset \\ 0 & , \text{ wenn } \omega = \omega' = \emptyset \\ C & \text{sonst.} \end{cases} \quad (5-2)$$

Der Parameter  $\Delta$  gewichtet den Fehler, wenn mehrere Trajektorien einer Menge einer Trajektorie der anderen Menge zugeordnet werden.

### 5.3.3 Intersection over Union

Granström et al.<sup>35</sup> verwendet zur Bewertung von Extended Object Trackern das Intersection-over-Union-Verfahren (IoU). Neben der Position der Objekte werden dabei auch ihre Abmessungen berücksichtigt. Die Metrik, auch Jaccard-Index genannt, ist als Verhältnis von Schnittmenge und Vereinigungsmenge zweier Mengen definiert.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5-3)$$

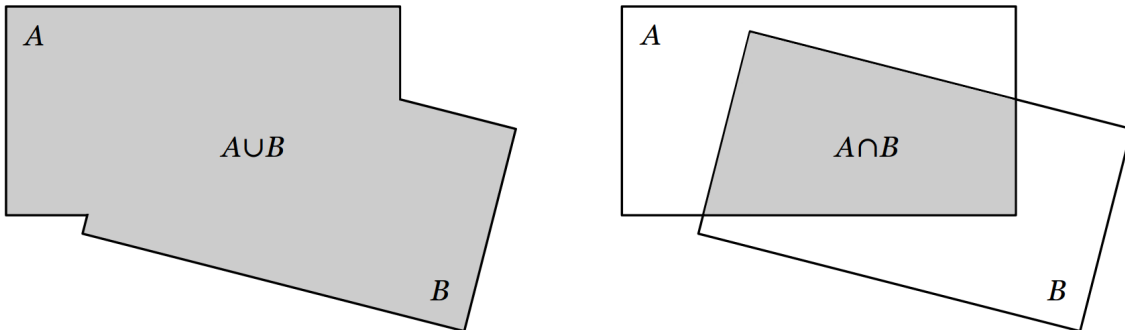


Abbildung 5-5: Schnittmenge und Vereinigungsmenge

<sup>34</sup> Vu, T.; Evans, R.: Optimal Subpattern Assignment Metric for Multiple Tracks (2018)

<sup>35</sup> Granström, K.; Baum, M.; Reuter, S.: Extended Object Tracking: Introduction, Overview and Applications (2017)

---

## 6 Ergebnisse

---

Im folgenden Kapitel werden die zuvor beschriebenen Metriken angewandt, um die Ergebnisse des Objektlistenmodells zu analysieren. Den Schwerpunkt bildet der Vergleich zwischen den Objektlisten des realen Sensors und des Ibeo-Modells. Anschließend werden die Objektlisten des Ibeo-Modells denen des erweiterten Modells sowie der idealen Objektliste gegenüber gestellt. Im Folgenden werden die aufgezeichneten Daten des realen Ibeo-Sensors als reale Daten bezeichnet, die Ground Truth Daten dagegen als ideale Daten/Objektlisten.

### 6.1 Vergleich von realen und simulierten Objektlisten

Im folgenden Abschnitt werden die Unterschiede zwischen den realen Daten und den Objektlisten des Ibeo-Modells untersucht.

#### 6.1.1 Versuch 1

Abbildung 6-1 stellt die vom realen und simulierten Sensor berechneten Trajektorien der Versuchsfahrzeuge in Szenario 1 dar. Der qualitative Vergleich zeigt, dass reale und simulierte Trajektorien gleichmäßig verlaufen und einen kleinen Offset zueinander aufweisen.

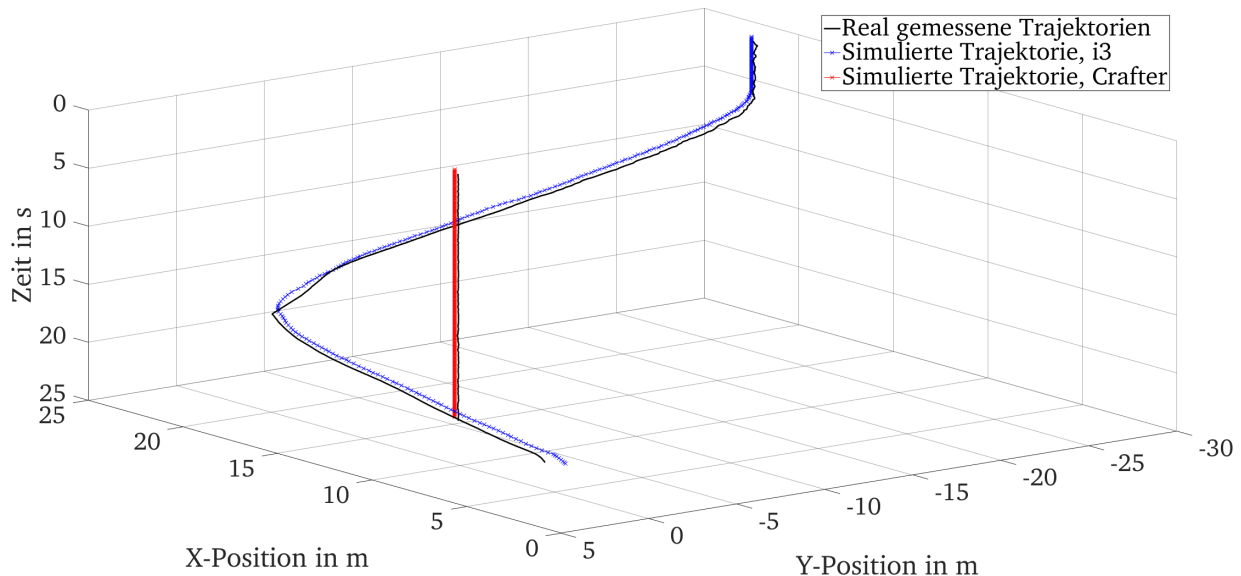


Abbildung 6-1: Reale und simulierte Trajektorien, Versuch 1

In Abbildung 6-3 ist der zeitliche Verlauf der Relativgeschwindigkeiten abgebildet, die Geschwindigkeiten der simulierten Objektlisten haben dabei einen kontinuierlicheren Verlauf als in der realen Messung. Die simulierte Relativgeschwindigkeit  $v_y$  des i3 bleibt ab Sekunde 14 länger bei ca.  $3 \frac{\text{m}}{\text{s}}$  als beim realen Sensor, was durch das verwendete Sensormodell erklärt werden kann. Ab diesem Zeitpunkt wird der i3 durch den Crafter verdeckt. Das Modell berücksichtigt nur den Hinweg der Lidarstrahlen, d.h. alles wird als vollständig reflektierend angenommen. Dadurch enthält die generierte

Punktwolke auch Punkte des verdeckten i3. Diese Punkte werden durch Strahlen erreicht, die unterhalb des Crafters verlaufen (vgl. Abbildung 6-2), der eine große Bodenhöhe besitzt. In der Realität werden diese Strahlen allerdings so reflektiert, dass sie am Empfänger nicht detektiert werden können und sind somit nicht in der segmentierten Punktwolke enthalten.



Abbildung 6-2: Unterhalb des Crafters verlaufende Lidarstrahlen

Außerdem kann es in der Simulation dazu kommen, dass Bodenreflexionen im Bereich der idealen Bounding Box fälschlicherweise der segmentierten Punktwolke zugeordnet werden. Somit wird ein Teil des i3 trotz Verdeckung weiterhin erfasst. Dies erklärt auch den gleichmäßigeren Verlauf der Geschwindigkeit in X-Richtung: Im Gegensatz zum realen Sensor wird in der Simulation der Abbiegevorgang des i3 trotz Verdeckung weiterhin wahrgenommen. Die simulierte Relativgeschwindigkeit liegt daher zwar näher an der Ground Truth, entspricht aber nicht dem realen Sensorverhalten.

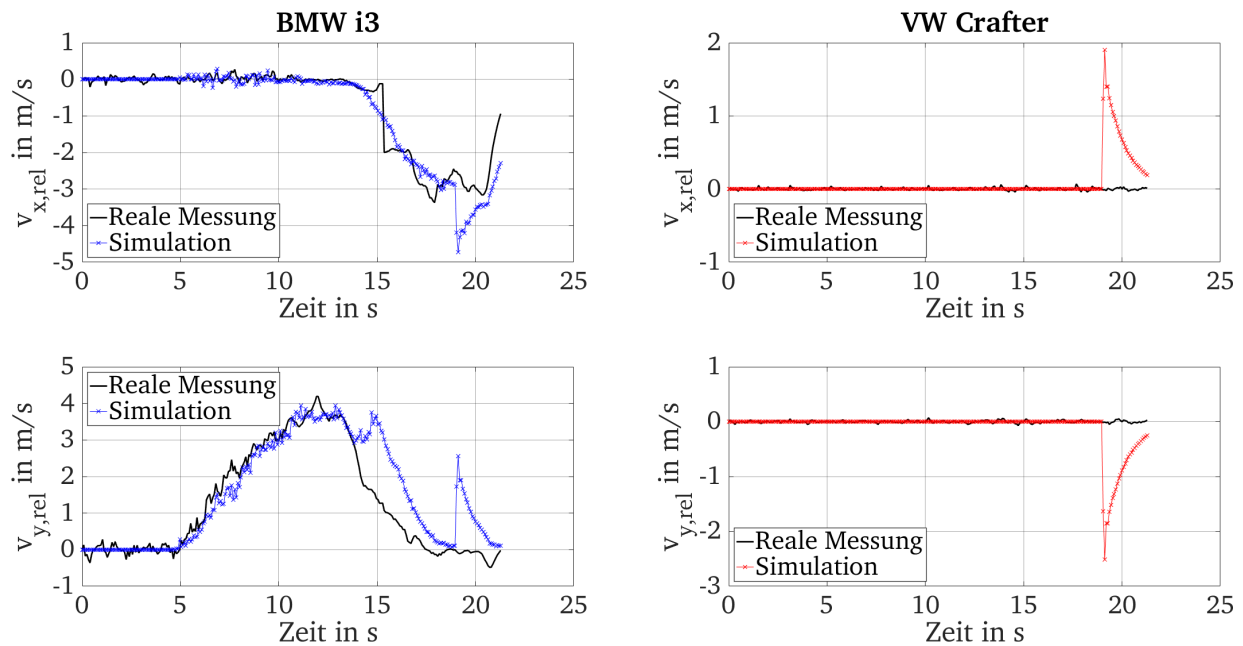


Abbildung 6-3: Reale und simulierte Relativgeschwindigkeiten, Versuch 1

Die simulierten Relativgeschwindigkeiten beider Fahrzeuge enthalten zudem bei ca. 35 s einen Sprung. Die Ursache hierfür liegt nicht am Objektlistenmodell, sondern an der Integration der Trajektorien in VTD. Wenn die Orientierung des i3 von  $180^\circ$  auf  $-180^\circ$  wechselt, kommt es in der Simulation zu einem kurzzeitigen Sprung der Orientierung. Durch die folglich stark veränderte Punktwolke kommt es zu einem Sprung in der Relativgeschwindigkeit. Dies erklärt aber nicht den Sprung beim Crafter, da weder beim Ego-Fahrzeug noch beim Crafter in der Simulation ein ähnliches Verhalten beobachtet werden kann. Dadurch, dass die Sprünge zum gleichen Zeitpunkt auftreten, wird die

Abweichung ebenfalls auf einen internen Fehler der Simulationsumgebung beim Simulieren der Trajektorien zurückgeführt.

Die Boxplots der Objektanmessungen in Abbildung 6-4 zeigen, dass sowohl der Median als auch die Streuung der Länge und Breite des i3 in den realen und simulierten Objektlisten nahezu identisch sind. Da sich Crafter und Ego-Fahrzeug im ersten Versuch nicht bewegen, ist die Streuung der berechneten Abmessungen sehr gering. Der Median der berechneten Länge des Crafters ist in der Simulation weitaus größer als beim realen Sensor. Dies hängt mit dem Punktwolkenmodell zusammen. Ego-Fahrzeug und Crafter sind gleich ausgerichtet, dadurch erreichen nur wenige Strahlen des Lidarsensors die Seite des Crafters. In der Realität werden die meisten Strahlen dort so reflektiert, dass sie bei den Empfängern der Sensoren nicht mehr ankommen, somit ist die wahrgenommene Länge nahezu Null. Da das Sensormodell nur den Hinweg der Strahlen berücksichtigt, enthält die segmentierte Punktwolke der Simulation dennoch Punkte von der Seite des Crafters. Dadurch wird die Länge des Crafters größer geschätzt und liegt näher an der Ground Truth, ist aber nicht realitätsnah. Da die Position des Fahrzeugs als geometrischer Schwerpunkt der segmentierten Punktwolke berechnet wird, führen die unterschiedlichen Punktwolken ebenfalls zu Abweichungen der Objektposition zwischen realen und simuliertem Sensorverhalten.

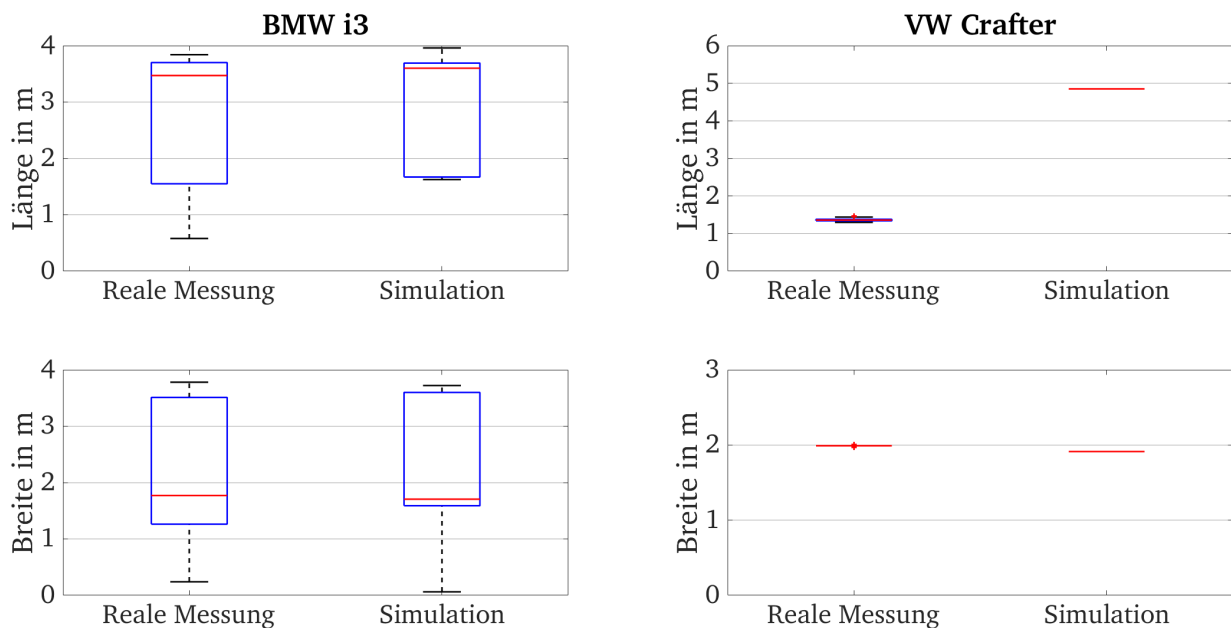


Abbildung 6-4: Reale und simulierte Objektanmessungen als Boxplots, Versuch 1

In Tabelle 6-1 ist der RMSE der Objektposition, der Relativgeschwindigkeiten in X- und Y-Richtung, sowie der Länge und Breite von beiden Fahrzeugen dargestellt. Da der Ibeo Lux 2010 wie in Kapitel 2 erläutert kaum Informationen über die Höhe von Objekten liefern kann, wird die Position ausschließlich in der X-/Y-Ebene betrachtet. Mit dem RMSE können die Abweichungen zwischen realen und simulierten Objektlisten quantitativ bewertet werden, allerdings werden die zeitlichen Verläufe der Abweichungen nicht berücksichtigt.

Tabelle 6-1: RMSE für reale Daten und Ibeo-Modell, Versuch 1

RMSE	BMW i3	VW Crafter
Geom. Mittelpunkt in m	0,681	0,694
Relativgeschwindigkeit $v_x$ in $\frac{m}{s}$	0,411	0,273
Relativgeschwindigkeit $v_y$ in $\frac{m}{s}$	0,726	0,359
Länge in m	0,801	3,488
Breite in m	0,498	0,078

Der RMSE der Trajektorie des BMW i3 beträgt 0,681 m und ist damit geringer als in der Mastertesis von Huch<sup>36</sup> mit den bisher am Fachgebiet FZD verwendeten Algorithmen zur Verarbeitung der simulierten Punktwolken. Der RMSE der Trajektorie des Crafters liegt mit 0,694 m in derselben Größenordnung. Der RMSE der Relativgeschwindigkeiten des Crafters beträgt 0,273  $\frac{m}{s}$  bzw. 0,359  $\frac{m}{s}$ . Die Abweichungen sind beim BMW i3 mit 0,411  $\frac{m}{s}$  und 0,726  $\frac{m}{s}$  deutlich größer, diese sind durch die zuvor beschriebenen unrealistischen simulierten Punktwolken während der Verdeckung zu erklären. Beim RMSE von Länge und Breite sind ebenfalls Unterschiede zwischen beiden Fahrzeugen festzustellen. Die Breite des Crafters ist in realer und simulierter Objektliste nahezu identisch. Der RMSE der Länge beträgt dagegen 3,488 m, dies ist auf die Ausrichtung des Crafters und die vernachlässigten Reflexionseigenschaften beim Sensormodell zurückzuführen. Der RMSE der Länge des i3 ist trotz der kurzzeitigen Verdeckung während des Abbiegevorgangs mit 0,801 m deutlich geringer als beim Crafter. Dies liegt daran, dass der i3 während des gesamten Versuchs so orientiert ist, dass die meisten Strahlen des realen Sensors an der Fahrzeugseite so reflektiert werden, dass sie den Empfänger wieder erreichen. Somit stimmt die real gemessene Punktwolke genauer mit der simulierten Punktwolke überein. Der höhere RMSE der Breite im Vergleich zum Crafter entsteht aus dem folglich schlechteren Blickwinkel auf die Fahrzeugfront und der Verdeckung während dem Abbiegen.

Um den zeitlichen Verlauf der Abweichung zwischen den realen und simulierten Objektlisten zu betrachten, wird im Folgenden die OSPA-MT-Metrik angewandt. Das Ergebnis der Metrik, der lokale Fehler und der Kardinalitätsfehler sind in Abbildung 6-5 in Abhängigkeit der Zeit dargestellt. Der Kardinalitätsfehler nimmt mit der Anzahl an Trajektorien einer Menge, die nicht einer Trajektorie der anderen Menge zugeordnet werden kann, zu. Das Ergebnis der Metrik wird dabei vom Kardinalitätsfehler dominiert, der lokale Fehler ist verhältnismäßig klein. Das kommt daher, dass die reale Objektliste viele False Positives enthält, der reale Tracker identifiziert bis zu 200 Objekte. Das Objektlistenmodell berücksichtigt dagegen keine Existenzunsicherheiten und gibt maximal zwei Objekte aus. Durch die unterschiedliche Trajektorienanzahl wird das Ergebnis der OSPA-MT-Metrik unbrauchbar. Für aussagekräftigere Ergebnisse werden die relevanten Trajektorien der realen Ibeo-Daten extrahiert und mit den simulierten Trajektorien verglichen. Das Ergebnis der OSPA-MT Metrik mit vernachlässigten False Positives ist in Abbildung 6-6 dargestellt.

<sup>36</sup> Huch, S.: Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation (2018) , S. 84

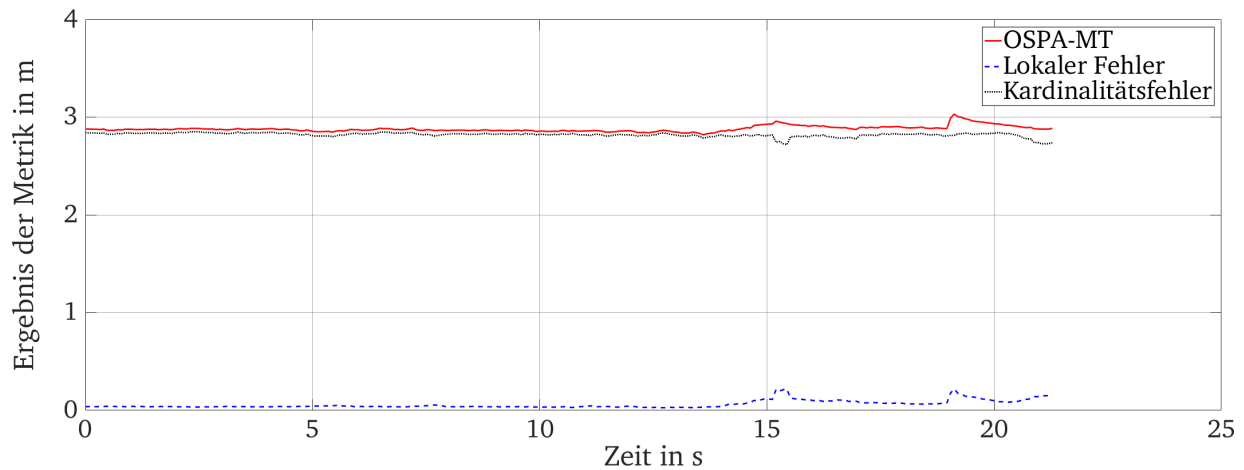


Abbildung 6-5: OSPA-MT Metrik für reale Daten und Ibeo-Modell, Versuch 1

Der Kardinalitätsfehler ist nun immer Null und das Ergebnis der OSPA-MT Metrik entspricht genau dem lokalen Fehler. Zu Beginn liegt der lokale Fehler unter 1 m und nimmt bis Sekunde 13 weiter ab. Anschließend nimmt der lokale Fehler zu und erreicht kurzzeitig einen Wert von 2,3 m. Der Anstieg entsteht durch die Abweichungen von Position und Relativgeschwindigkeiten durch die zuvor beschriebene Berücksichtigung von unterschiedlichen segmentierten Punktwolken während der Verdeckung des i3. Bei ca. 19 Sekunden steigt der lokale Fehler auf über 3 m, bevor er wieder auf 1,5 m fällt. Der zweite Peak resultiert aus den Abweichungen der Relativgeschwindigkeiten beim Übergang der Orientierung des i3 von  $180^\circ$  zu  $-180^\circ$  (vgl. Abbildung 6-3).



Abbildung 6-6: OSPA-MT Metrik für relevante reale Daten und Ibeo-Modell, Versuch 1

Anschließend wird mit der IoU-Metrik die Übereinstimmung der Bounding Boxes bewertet, Abbildung 6-7 zeigt den zeitlichen Verlauf der IoU Metrik. Die IoU-Werte des Crafters bleiben während des gesamten Versuchs relativ konstant bei 0,2. Die IoU-Werte des BMW i3 sind zu Beginn des Versuchs bei 0,3 und gehen während der i3 verdeckt wird gegen Null. Danach steigt die Übereinstimmung zwischen realen und simulierten Bounding Boxes auf bis zu 0,6, bevor der i3 das Egofahrzeug passiert und der Wert auf Null sinkt.

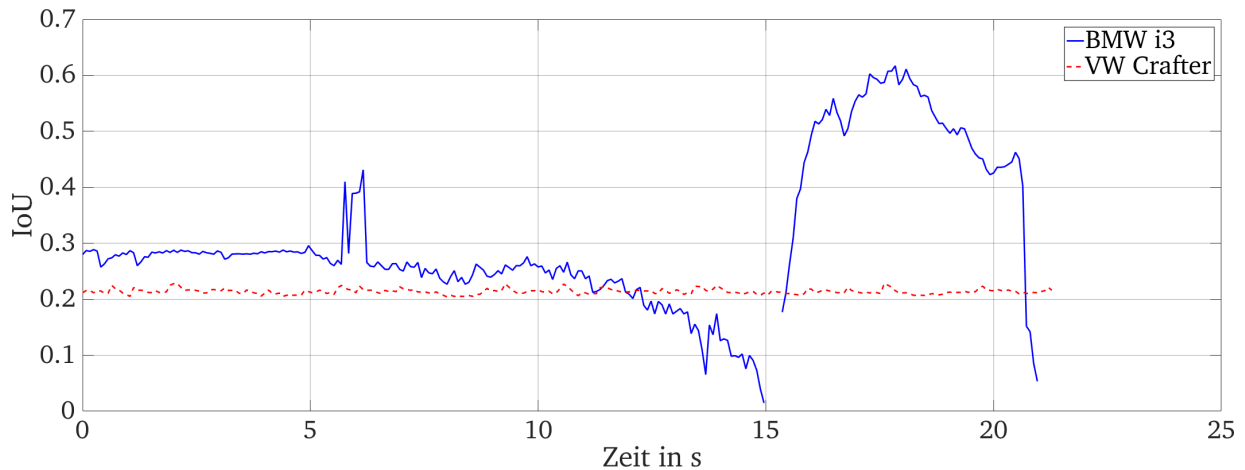


Abbildung 6-7: IoU für reale Daten und Ibeo-Modell, Versuch 1

### 6.1.2 Versuch 2

Im folgenden Abschnitt werden die gleichen Metriken auf die Daten aus Versuch 2 angewandt. Abbildung 6-8 zeigt den zeitlichen Verlauf der vom realen sowie simulierten Sensor berechneten Positionen der Fahrzeuge. Wie im ersten Versuch liegen die Verläufe von realen und simulierten Trajektorien nahe beieinander, nur während der i3 abbiegt und vom Crafter verdeckt wird nimmt die Abweichung zu.

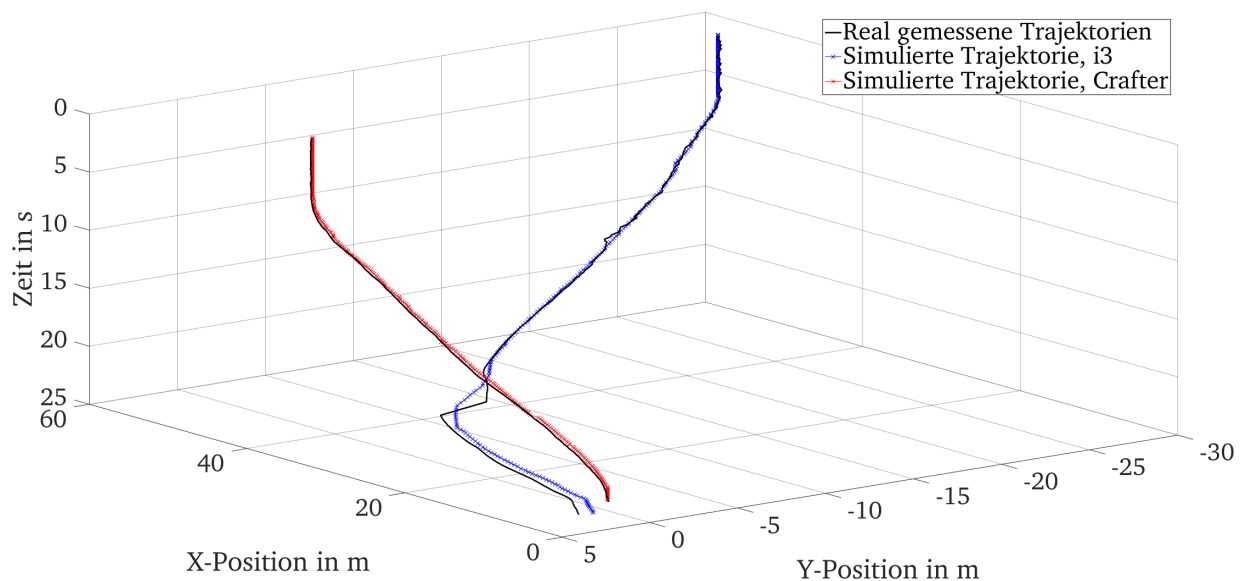


Abbildung 6-8: Reale und simulierte Trajektorien, Versuch 2

Abbildung 6-9 zeigt den Verlauf der Relativgeschwindigkeiten im zweiten Versuch. Bei Änderungen der Beschleunigung von i3 und Ego-Fahrzeug weichen die Verläufe der Realität und Simulation leicht voneinander ab. Wie in Versuch 1 ist die Geschwindigkeit des i3 in Y-Richtung aufgrund der Bodenreflexionen länger konstant. Auch in diesem Versuch ist ein deutlicher Ausreißer in den Relativgeschwindigkeiten beider Fahrzeuge bei ca. 35 Sekunden zu beobachten, beim Wechsel der Orientierung von  $180^\circ$  auf  $-180^\circ$  des i3 auftritt.

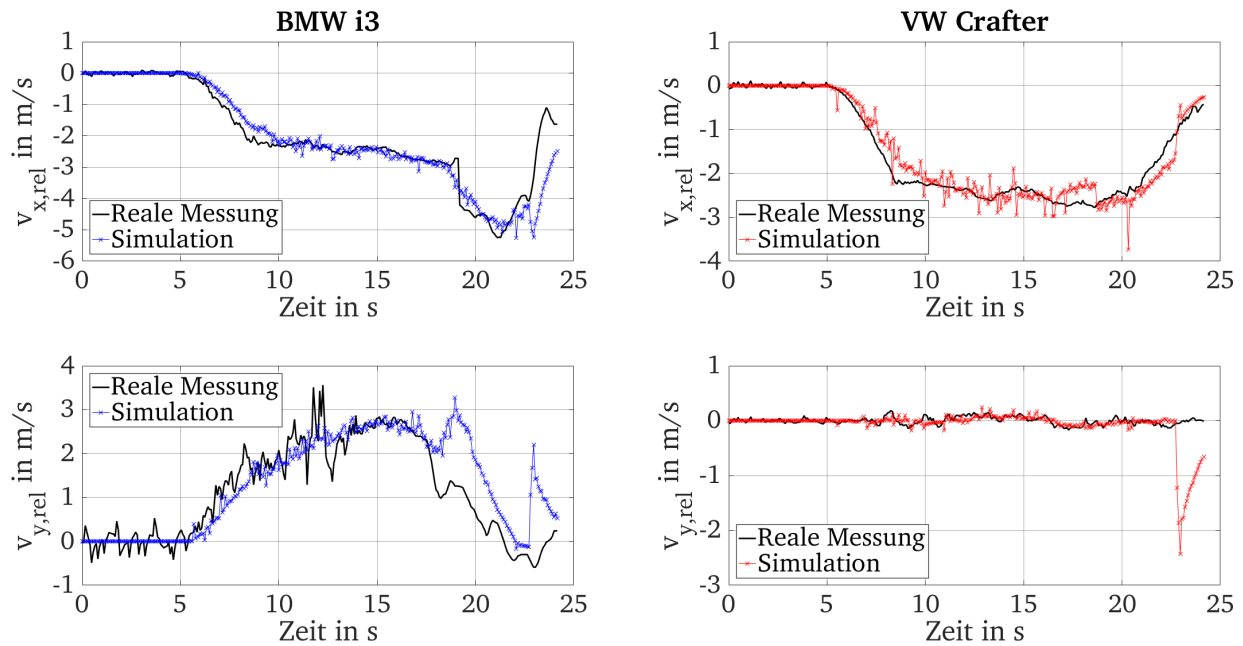


Abbildung 6-9: Reale und simulierte Relativgeschwindigkeiten, Versuch 2

In Abbildung 6-10 sind die berechneten Abmessungen der beiden Fahrzeuge als Boxplots abgebildet. Die Streuung der Abmessungen des Crafters ist größer als im ersten Versuch, da sich das Ego-Fahrzeug auf den Crafter zubewegt und dadurch die simulierte Punktwolke stärker variiert. Der Median der Breite ist in der simulierten und realen Objektliste etwa gleich groß. Der Median der simulierten Länge des Crafters ist analog zum ersten Versuch größer als in der realen Messung, was aus der Ray-Casting-Methode des Sensormodells resultiert. Der Median der Länge des i3 ist in der simulierten Objektliste deutlich höher als in der realen Objektliste, gleichzeitig ist der Median der Länge in der Simulation niedriger als in der realen Messung. Dabei liegt der Median der real gemessenen Breite höher als die Ground Truth (1,79 m), was darauf hindeutet, dass der reale Sensor die Orientierung des i3 erst später korrekt bestimmt und somit die Länge und Breite länger falsch zuordnet.

In Tabelle 6-2 ist der RMSE für die Position, Relativgeschwindigkeiten, Länge und Breite der beiden Fahrzeuge aufgelistet. Der RMSE des geometrischen Mittelpunktes ist mit 0,520 m und 0,354 m geringer als im ersten Versuch. Der RMSE der Relativgeschwindigkeiten liegt in der gleichen Größenordnung wie im ersten Versuch. Die unterschiedlichen Boxplots der Abmessungen des i3 führen auch zu einem höheren RMSE. Der RMSE der Abmessungen des Crafters liegt mit 3,898 m und 0,100 m liegt im selben Bereich wie im ersten Versuch.

Tabelle 6-2: RMSE für reale Daten und Ibeo-Modell, Versuch 2

RMSE	BMW i3	VW Crafter
Geom. Mittelpunkt in m	0,520	0,354
Relativgeschwindigkeit $v_x$ in $\frac{m}{s}$	0,514	0,267
Relativgeschwindigkeit $v_y$ in $\frac{m}{s}$	0,656	0,334
Länge in m	1,311	3,898
Breite in m	1,106	0,100



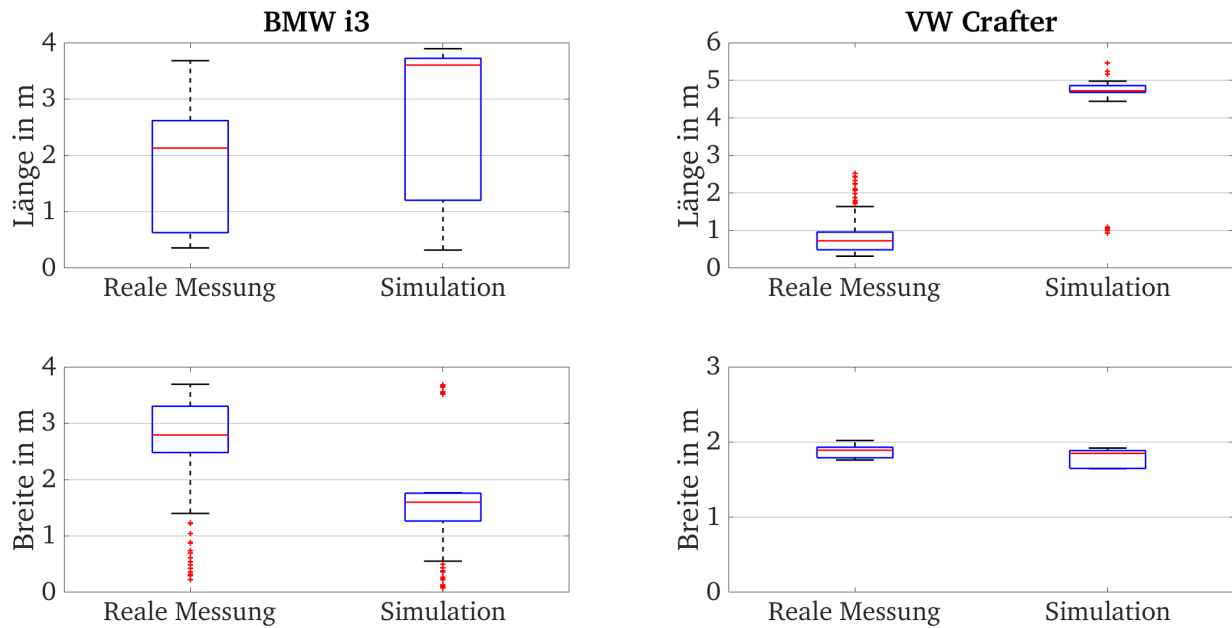


Abbildung 6-10: Reale und simulierte Objektanmessungen als Boxplots, Versuch 2

In Abbildung 6-11 ist das Ergebnis der OSPA-MT-Metrik zwischen den simulierten und den relevanten realen Trajektorien in Abhängigkeit der Zeit abgebildet. Bis Sekunde 17 liegt der lokale Fehler bis auf einzelne Ausreißer unterhalb von 0,5 m, anschließend steigt der Fehler aufgrund der Verdeckung des i3 auf über ca. 2 m an. Bei ca. 23 Sekunden hat der Fehler mit 3 m einen Peak, der wieder durch die Abweichung der Relativgeschwindigkeiten aufgrund des Sprungs der Orientierung beim i3 entsteht.

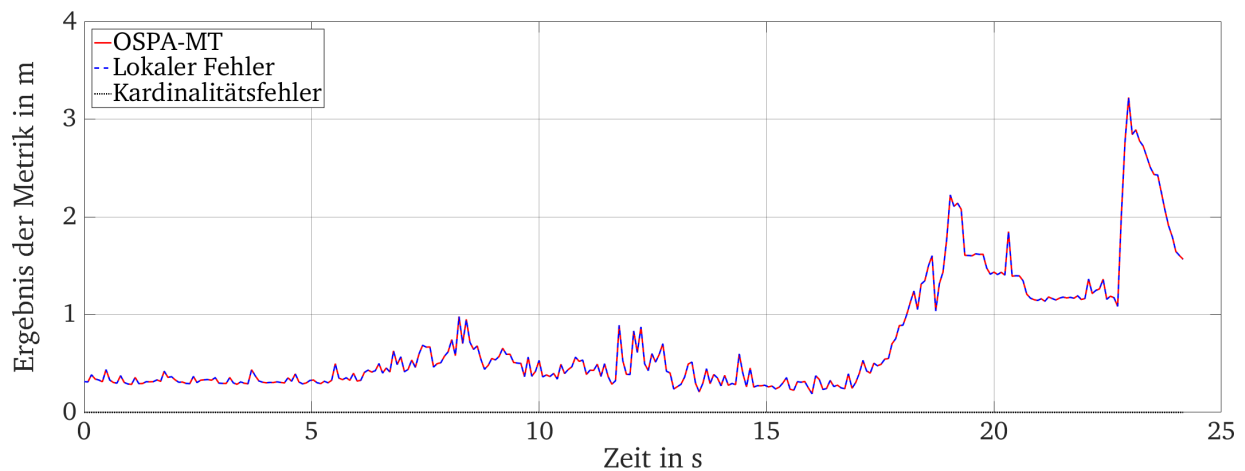


Abbildung 6-11: OSPA-MT Metrik für relevante reale Daten und Ibeo-Modell, Versuch 2

Abbildung 6-12 zeigt die Ergebnisse der IoU-Metrik. Die Übereinstimmung der realen und simulierten Bounding Boxes des Craftes liegt die meiste Zeit zwischen 0,1 und 0,2, gegen Ende der Messung steigt der IoU-Wert bis auf 0,4 an. Die Übereinstimmung der Bounding Boxes des i3 liegen zunächst bei 0,1, nachdem beide Fahrzeuge ab Sekunde 6 beschleunigen steigt der Wert bis auf 0,6. Während der Verdeckung des i3 zwischen Sekunde 17 und 20 geht der IoU-Wert auf 0 zurück. Ab Sekunde 23, wo sich der i3 am Rand des linken Sensorsichtfeldes befindet, geht der IoU-Wert auf 0 zurück.

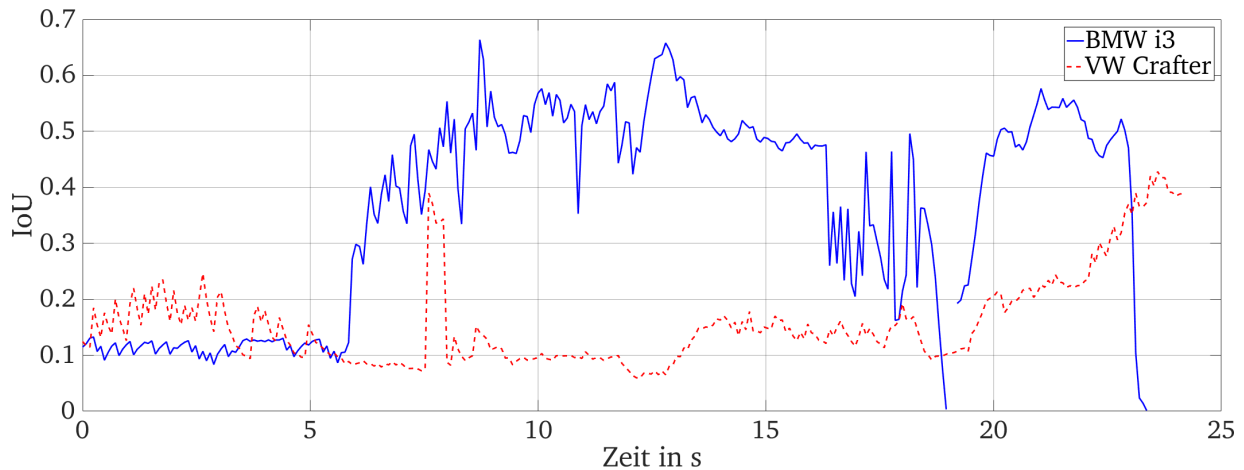


Abbildung 6-12: IoU für reale Daten und Ibeo-Modell, Versuch 2

Insgesamt wird festgehalten, dass das implementierte Objektlistenmodell Ergebnisse liefert, die bereits sehr ähnlich zu den Objektlisten des realen Sensors sind. Der Hauptgrund für die bestehende Abweichung liegt am verwendeten Sensormodell, das die synthetischen Punktwolken liefert. Wird dieses Modell so erweitert, dass Bodenreflexionen gefiltert werden und die Reflexion an Zielobjekten berücksichtigt wird, ist eine deutliche Annäherung von realen und simulierten Objektlisten zu erwarten. Alternativ kann eine Filterung der Bodenreflexionen sowie die Berücksichtigung der Reflexionseigenschaften von Objekten auch im Objektlistenmodell implementiert werden.

## 6.2 Vergleich der simulierten Objektlisten aus Ibeo-Modell und erweitertem Modell

Nachdem im vorherigen Abschnitt die realen Objektlisten mit den simulierten Objektlisten des Ibeo-Modells verglichen wurden, sollen im Folgenden die Ergebnisse des erweiterten Ibeo-Modells evaluiert werden. Dazu werden die Objektlisten des erweiterten Modells mit denen des Ibeo-Modells und des idealen Modells im zweiten Versuch verglichen. Die Ergebnisse des ersten Versuchs sind im Anhang zu finden. Um eine gegenseitige Beeinflussung der Funktionen auszuschließen, werden sie einzeln simuliert und ausgewertet.

### 6.2.1 Erweitertes Modell mit verbesserter Positionsbestimmung

Zuerst wird die verbesserte Positionsbestimmung analysiert. Abbildung 6-13 zeigt die zeitlichen Verläufe der Positionen beider Fahrzeuge in X- und Y-Richtung. Die Trajektorien des erweiterten Modells liegen bei qualitativer Betrachtung deutlich näher an denen des idealen Modells als die Trajektorien des Ibeo-Modells.

Die berechneten RMSE-Werte in Tabelle 6-3 bestätigen den Eindruck. Der RMSE des verbesserten Modells ist mit 0,356 m beim i3 und mit 0,472 m beim Crafter deutlich niedriger als beim Ibeo-Modell.

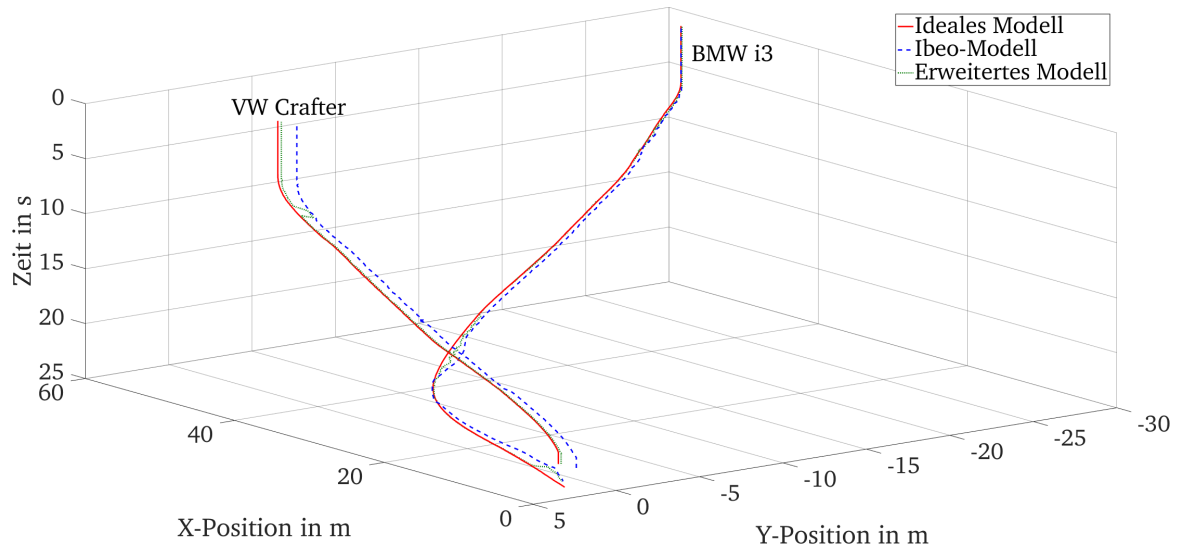


Abbildung 6-13: Trajektorien von idealem Modell, Ibeo-Modell und verbessertem Modell, Versuch 2

Tabelle 6-3: RMSE für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
Geom. Mittelpunkt in m	0,733	0,356	2,268	0,472

Wie der zeitliche Verlauf der Abweichungen von Position und Relativgeschwindigkeiten ist, zeigt sich an den Ergebnissen der OSPA-MT-Metrik in Abbildung 6-14. Bis auf vereinzelte Ausreißer ist das Ergebnis der Metrik beim erweiterten Modell während des gesamten Versuchs deutlich niedriger als beim Ibeo-Modell.

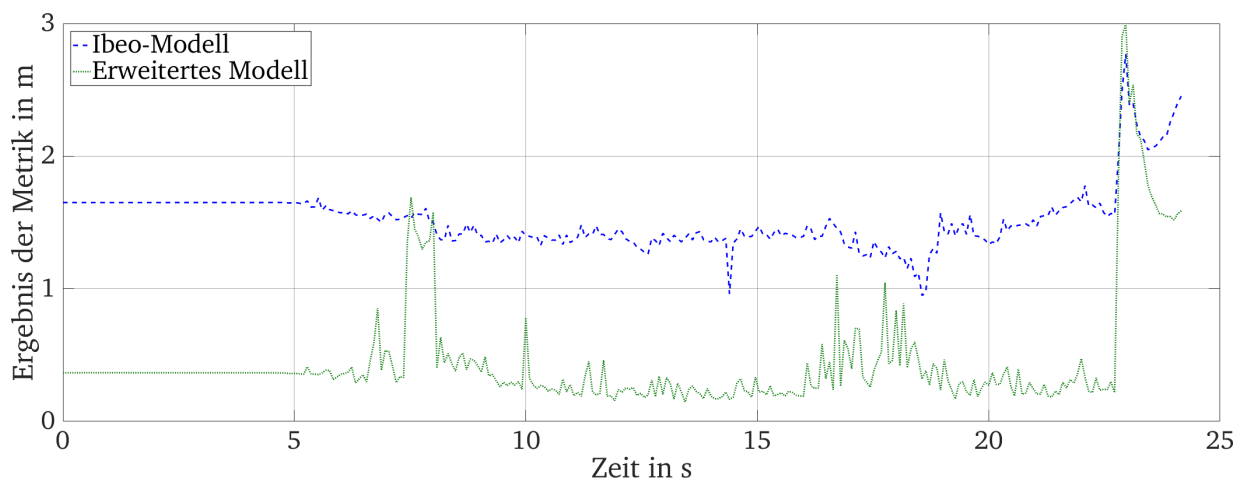


Abbildung 6-14: OSPA-MT-Metrik für Ibeo-Modell und erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2

Ähnlich verhält es sich bei der IoU-Metrik in Abbildung 6-15. Die Überschneidung der vom Modell bestimmten Bounding Box mit der Ground Truth Bounding Box ist zu fast jedem Zeitpunkt deutlich größer als beim Ibeo-Modell. Zeitweise erreicht das erweiterte Modell Spitzenwerte von 0,9. Ledig-

lich wenn die Abmessungen des Objektes unzureichend bestimmt werden - wie z.B. während der Verdeckung des i3 - fällt der IoU-Wert ab, bleibt aber dennoch über denen des Ibeo-Modells.

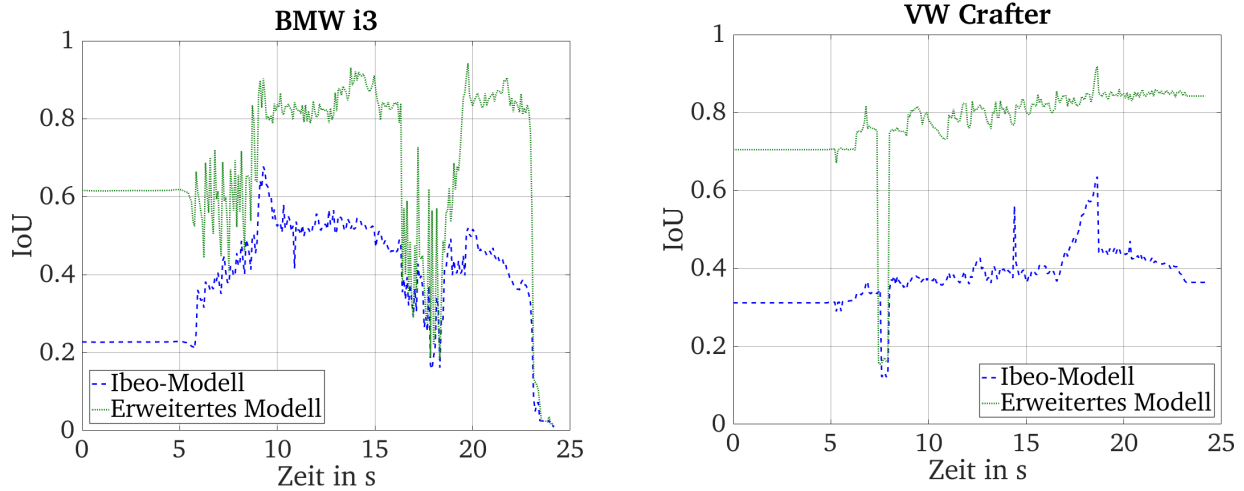


Abbildung 6-15: IoU für Ibeo-Modell und erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 2

Die verbesserte Positionsbestimmung liefert somit in allen Belangen deutlich bessere Ergebnisse als das Ibeo-Modell und ist zur Simulation modernerer Sensoren geeignet.

## 6.2.2 Erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung

Im folgenden Abschnitt werden die Ergebnisse des erweiterten Modells mit verbesserter Geschwindigkeitsbestimmung analysiert. In Abbildung 6-16 sind die zeitlichen Verläufe der Relativgeschwindigkeiten dargestellt.

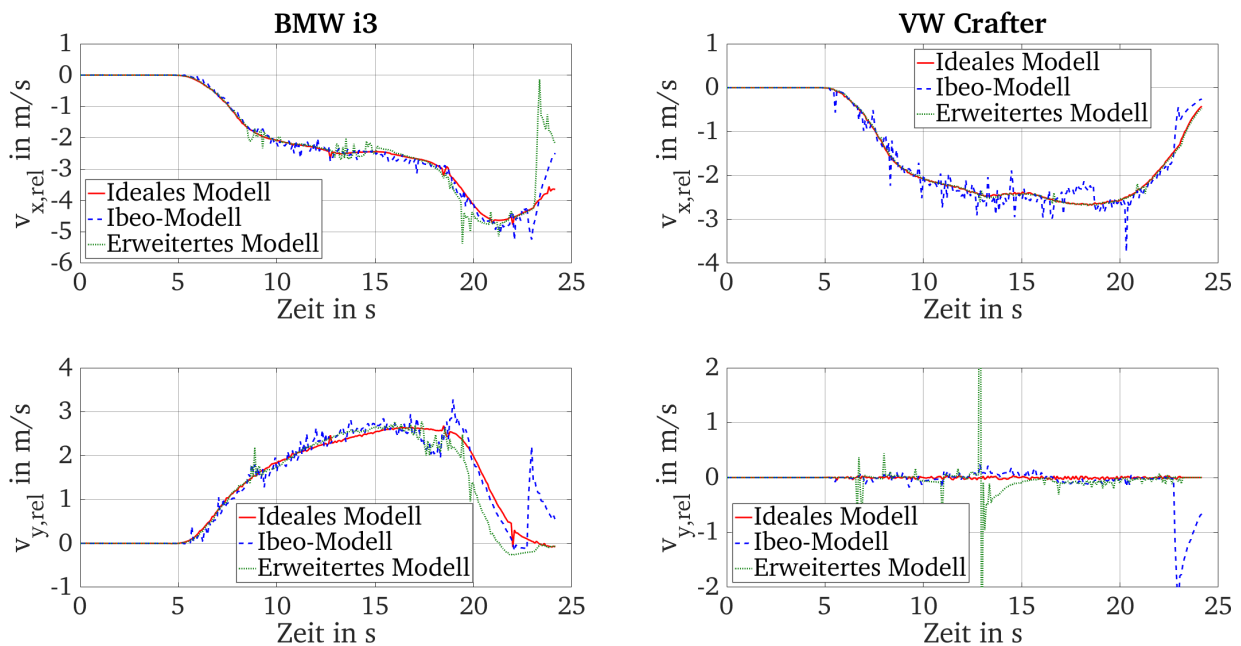


Abbildung 6-16: Relativgeschwindigkeiten von idealem Modell, Ibeo-Modell und erweitertem Modell, Versuch 2

Die berechneten Relativgeschwindigkeiten des erweiterten Modells verlaufen zeitweise gleichmäßiger und liegen damit oft näher an der Ground Truth als die des Ibeo-Modells, weisen aber einige Ausreißer auf. Durch die Mittelung der Geschwindigkeit werden die maximalen Abweichungen zwar verringert, sehr große Ausreißer können aber zu Abweichungen führen, die über einen längeren Zeitraum abgebaut werden müssen (siehe Diagramm rechts unten in Abbildung 6-16, Sekunde 13-16).

Die Ausreißer wirken sich auch auf den RMSE des erweiterten Modells in Tabelle 6-4 aus. Während der RMSE von  $v_{x,rel}$  des Crafters und  $v_{y,rel}$  des i3 beim erweiterten Modell kleiner sind, ist der Fehler bei  $v_{y,rel}$  des Crafters und  $v_{x,rel}$  des i3 deutlich größer als beim Ibeo-Modell. Ein möglicher Grund für die Ausreißer ist ein Wechsel des Referenzpunktes, der im erweiterten Modell bisher nicht berücksichtigt wird. Wechselt der Referenzpunkt aufgrund der Position und Orientierung des Objektes auf eine andere Ecke, entstehen in der Ableitung des Referenzpunktes aufgrund des kleinen Zeitschrittes große Sprünge.

Tabelle 6-4: RMSE für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 2

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
$v_x$ in $\frac{m}{s}$	0,207	0,503	0,211	0,033
$v_y$ in $\frac{m}{s}$	0,331	0,293	0,334	0,634

Gleiches zeigt der zeitliche Verlauf des Ergebnisses der OSPA-MT-Metrik in Abbildung 6-17. Die OSPA-MT-Distanz des erweiterten Modells ist nur zeitweise niedriger als die des Ibeo-Modells. Zu den Zeitpunkten, an denen in den Relativgeschwindigkeiten die zuvor beobachteten Sprünge auftreten (Sekunde 13, 19-23 sowie 24), ist die OSPA-MT-Distanz des erweiterten Modells erkennbar größer.

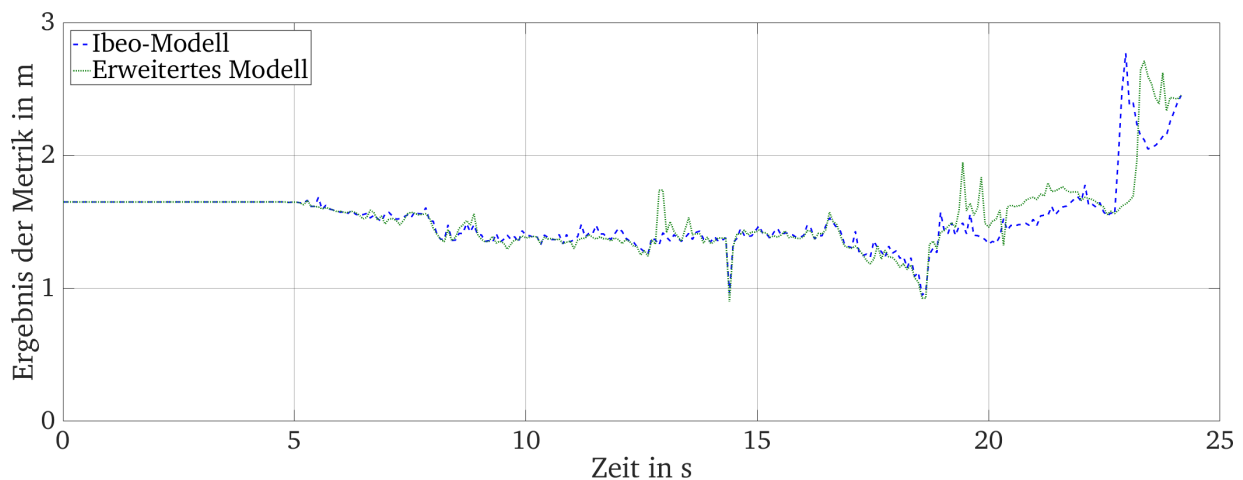


Abbildung 6-17: OSPA-MT-Metrik für Ibeo-Modell und erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 2

Insgesamt bietet die aktuelle Implementierung zur verbesserten Geschwindigkeitsberechnung noch keinen Mehrwert gegenüber dem Ibeo-Modell. Dennoch lohnt es sich zu evaluieren, ob eine Erkennung von Sprüngen des Referenzpunktes sowie eine geringere Mittelung der berechneten Werte zu geringeren Abweichungen führt.

### 6.2.3 Erweitertes Modell mit verbesserter Bestimmung der Objektabmessungen

Abbildung 6-18 stellt die zeitlichen Verläufe der Länge und Breite beider Fahrzeuge dar. Das erweiterte Modell „merkt“ sich Abmessungen, die bei einer vorteilhaften Position und Orientierung des Objektes bestimmt werden. Dieser Wert kann allerdings auch niedriger als beim Ibeo-Modell liegen. Die vom Ibeo-Modell berechnete Breite des i3 liegt beispielsweise zwischen Sekunde 9 und 16 sehr nahe an der Ground Truth, obwohl der i3 quer zum Ego-Fahrzeug ausgerichtet ist. Daher ist der Qualitätswert der Breite gering und das erweiterte Modell übernimmt dort nicht den Wert des Ibeo-Modells, sondern erst ab Sekunde 20, nachdem der i3 abgebogen ist. Auffällig ist zudem, dass vor allem die berechnete Länge und Breite des i3 zwischen Sekunde 16 und 20 trotz Verdeckung durch den Crafter nur leicht abfällt. Dies ist auf die unterschiedlichen Punktwolken zurückzuführen, die direkte Treffer der simulierten Lidar-Strahlen unterhalb des Crafters und Bodenreflexionen entstehen.

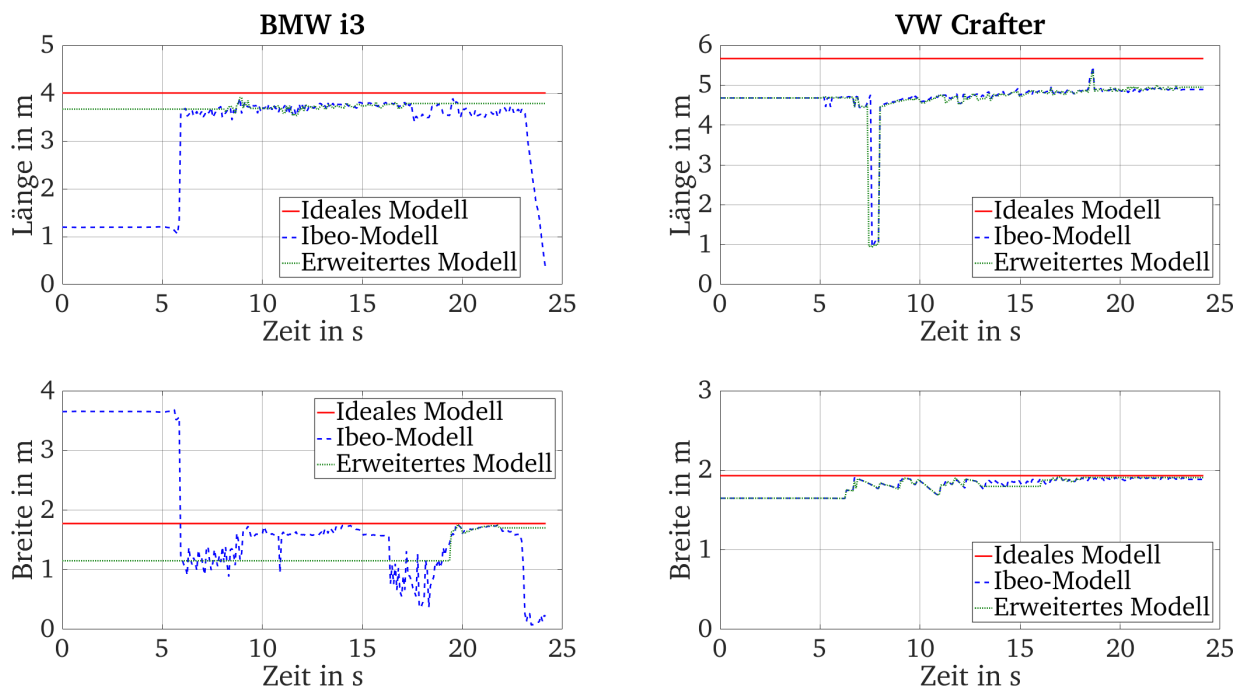


Abbildung 6-18: Objektabmessungen von Ibeo-Modell und verbessertem Modell, Versuch 2

Die Boxplots der Objektabmessungen in Abbildung 6-19 zeigen, dass sich die Streuung der Werte reduziert. Wie die Breite des i3 zeigt, kann der Median Werte annehmen, die weiter von der Ground Truth entfernt liegen als das Ibeo-Modell. Gleichzeitig treten dafür keine Ausreißer mehr nach unten auf.

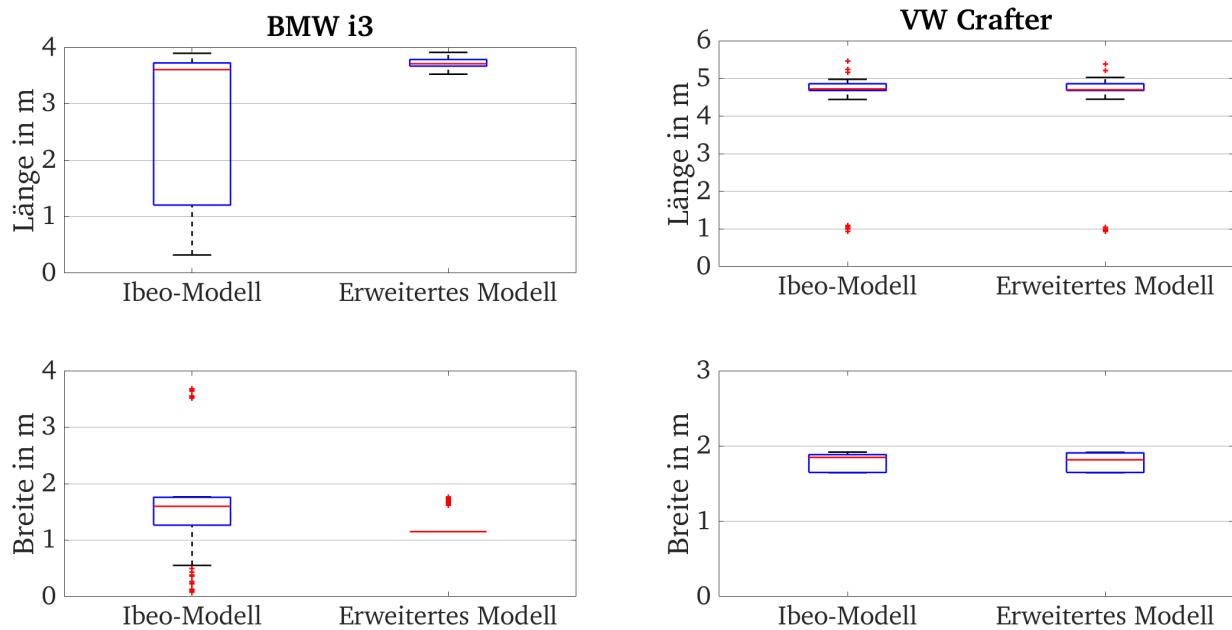


Abbildung 6-19: Objektanmessungen von Ibeo-Modell und verbessertem Modell als Boxplots, Versuch 2

Da die Sensorunsicherheit zur Zuordnung der Länge und Breite bei nicht eindeutig bestimmten Orientierungen des Objektes nur im Ibeo-Modell implementiert ist, liegen die Werte der beiden Modelle zu Beginn des Versuchs weit auseinander. Diese würde die Bewertung des erweiterten Modells verfälschen, daher wird der RMSE erst ab dem Zeitpunkt berechnet, ab dem auch beim Ibeo-Modell die Abmessungen korrekt zugeordnet werden. Die Ergebnisse sind in Tabelle 6-5 abgebildet. Bei der Länge des i3 ist die Abweichung zur Ground Truth mit dem erweiterten Modell geringer als mit dem Ibeo-Modell. Obwohl die Breite des i3 mit dem Ibeo-Modell teilweise deutlich näher an der Ground Truth liegt, ist der RMSE des erweiterten Modells mit 0,537 m trotzdem niedriger. Dies liegt einerseits daran, dass das Modell die teilweise Verdeckung des i3 von Sekunde 16-20 erkennt und die Breite aus der aktuellen Messung nicht übernommen wird. Andererseits registriert das erweiterte Modell gegen Ende der Messung ebenfalls, dass der i3 das Sichtfeld des Sensors verlässt und behält die zuvor registrierte Breite im Speicher. Der RMSE der berechneten Abmessungen des Crafters ist mit dem erweiterten Modell leicht höher als mit dem Ibeo-Modell.

Tabelle 6-5: RMSE für erweitertes Modell mit verbesserter Bestimmung der Objektanmessungen, Versuch 2

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
Länge in m	0,666	0,276	1,121	1,211
Breite in m	0,618	0,537	0,097	0,102

In Abbildung 6-20 sind die Ergebnisse der IoU-Metrik dargestellt. Da die Abmessungen des i3 beim erweiterten Modell wie zuvor beschrieben nicht immer näher an der Ground Truth als das Ibeo-Modell liegen, ist auch der IoU-Wert des erweiterten Modells teilweise niedriger als beim Ibeo-Modell. Dafür bleibt der IoU-Wert beim erweiterten Modell während der Verdeckung des i3 auf einem höheren Niveau als das Ibeo-Modell.

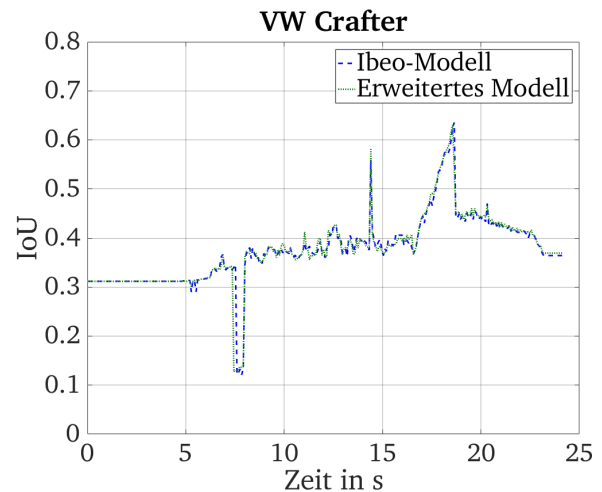
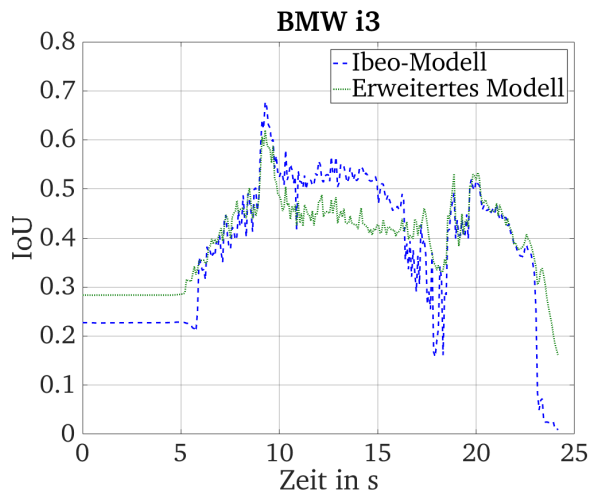


Abbildung 6-20: IoU für Ibeo-Modell und erweitertes Modell mit verbesserter Bestimmung der Objektanmessungen, Versuch 2

Das erweiterte Modell bietet die Möglichkeit, die in vergangenen Zeitschritten berechneten Abmessungen zu berücksichtigen. Mit dieser Funktion kann vermieden werden, dass die Informationen über die Größe eines Objekts verloren gehen, wenn die Abmessungen wegen der aktuellen Ausrichtung oder Position (Verdeckung, Distanz, Sichtfeldrand) des Objekts zu einem zukünftigen Zeitpunkt ungenauer bestimmt werden können. Gleichzeitig kann es in bestimmten Situationen vorkommen, dass die Abmessungen weiter von der Ground Truth abweichen als beim Ibeo-Modell. Durch eine Anpassung des Qualitätswertes kann die Funktion weiter optimiert werden, sodass das Modell bereits früher gut bestimmte Werte des Ibeo-Modells übernimmt. Außerdem ist anzumerken, dass die Funktion starke Auswirkungen auf die Bestimmung der Objektposition hat, falls die verbesserte Positionsbestimmung verwendet wird. Wenn der Referenzpunkt unzureichend bestimmt wird, z.B. aufgrund von Verdeckung, kann die Position weiter von der Ground Truth entfernt liegen als das Ibeo-Modell.



---

## 7 Diskussion der Ergebnisse

---

In diesem Kapitel werden die Ergebnisse dieser Arbeit kritisch diskutiert. Dazu wird der Einfluss von Fehlern bei der Methodik, mit der vorgegangen wurde, der Simulationsumgebung und des Sensormodells betrachtet.

### 7.1 Einfluss der Methodik

Bei der Aufzeichnung der realen Daten treten Messunsicherheiten auf, Ungenauigkeiten der GPS-Messung haben beispielsweise große Auswirkungen auf die Ergebnisse, da sie zu Abweichungen zwischen den realen und simulierten Trajektorien der Fahrzeuge führen.

Außerdem führt die instabile WLAN-Verbindung zur Übertragung der GPS-Daten zu Sprüngen in der gemessenen Position und Orientierung. Bei der Konvertierung der GPS-Daten in Trajektorien für VTD werden diese Sprünge detektiert und durch interpolierte Werte ersetzt, um die Abweichung zur realen Trajektorie gering zu halten. Dennoch wurden in jeder Simulation Sprünge in der Orientierung festgestellt. Diese treten am Ende des Abbiegevorangs auf, wenn die Orientierung von  $180^\circ$  auf  $-180^\circ$  springt. Dieser Sprung hatte vor allem Auswirkungen auf die ermittelte Relativgeschwindigkeit.

Ein weiterer Fehler entsteht dadurch, dass die Zeitstempel von GPS- und Sensordaten nicht identisch sind und die Daten manuell synchronisiert werden mussten.

Da nur zwei Messgeräte zur Aufzeichnung der GPS-Daten am Fachgebiet zur Verfügung stehen, mussten Position und Orientierung des statischen Crafters mit Maßband und Kompass bestimmt werden, das Messungenauigkeiten unterliegt. Die Messungenauigkeit der ADMA beträgt ca. 2 cm, der Fehler bei der Ausmessung wird auf 5 cm geschätzt.

Eine weitere Unsicherheit besteht bei der genauen Position und Ausrichtung der Lidarsensoren am Versuchsfahrzeug. Wie in Kapitel 5 beschrieben, wurde vor den Fahrversuchen eine Kalibrierung durchgeführt, die ebenfalls Messungenauigkeiten unterliegt. Des Weiteren wurde die Kalibrierung nur für den Gierwinkel und nicht für Nick- und Wankwinkel durchgeführt.

Einfluss auf die Ergebnisse hat außerdem die Wahl der Parameter für die OSPA-MT Metrik. Da kaum Erfahrungswerte für eine geeignete Parameterwahl existieren, wurden die von den Autoren empfohlenen Werte verwendet. Da die vielen False Positives der realen Objektliste das Ergebnis der Metrik unbrauchbar gemacht haben und daher aus der Objektliste entfernt wurden, ist der Cut-Off-Parameter aber nahezu bedeutungslos.

### 7.2 Einfluss der Simulationsumgebung

Bei der verwendeten Simulationsumgebung VTD gibt es verschiedene Limitierungen. Einerseits existiert nur eine begrenzte Auswahl an Fahrzeugmodellen für die Simulation. Für den Crafter musste ersatzweise ein Modell eines Sprinters von Mercedes-Benz verwendet werden, dessen geometrische Abmessungen dem Crafter ähnlich sind, aber dennoch von diesen abweichen.

---

Weiterhin kann das Objektlistenmodell in der Simulation aktuell nur mit einer Frequenz von 60 Hz ausgeführt werden. Der reale Sensor wie auch das Sensormodell wird allerdings mit einer Frequenz von 12,5 Hz betrieben. Das bedeutet, dass die Zeitstempel der Ground Truth Daten und der Punktwolken nie exakt zusammen passen. Dies wurde im Programm so gelöst, dass die Daten zum nächstmöglichen Zeitpunkt ausgelesen werden, nachdem das Sensormodell eine neue Punktwolke liefert, das heißt alle 0,08 Sekunden. Der maximale zeitliche Abstand zwischen Ground Truth Daten und Punktwolken beträgt dadurch bis zu 0,01666 Sekunden.

### 7.3 Einfluss des Sensormodells

Da für das implementierte Objektlistenmodell in vielen Arbeitsschritten die Punktwolke des physikalischen Sensormodells verwendet wird, sind die Ergebnisse maßgeblich von der Qualität des Sensormodells abhängig. Die beim Sensormodell getroffenen Vereinfachungen führen zu Abweichungen zwischen realer und simulierter Punktwolke. Den größten Einfluss hat die Eigenschaft, dass in der Simulation nur der Hinweg der Laserstrahlen berücksichtigt wird. Lidarstrahlen, die unter dem Crafter verlaufen führen zu einer Verfälschung der Punktwolke, die sich in dieser Masterthesis insbesondere während der Verdeckung des i3 auf alle ermittelten Eigenschaften ausgewirkt hat. Außerdem werden die Reflexionseigenschaften der Objektoberfläche vernachlässigt. Wo in der Realität die Strahlen zumeist so reflektiert werden, dass sie nicht mehr den Empfänger des Sensors erreichen, werden mit dem Sensormodell dennoch Punkte der Punktwolke hinzugefügt. Dies führt zu erheblichen Abweichungen bei der Berechnung der Position und den Abmessungen des Objekts.

Zudem wird mit dem Modell die Strahlaufweitung nicht berücksichtigt, in der Simulation sind die Lidar-Pulse punktförmig dargestellt. In der Realität nimmt dagegen der Durchmesser der Pulse mit der Entfernung zum Zielobjekt zu.

Ebenfalls nicht implementiert ist die Transmission der Atmosphäre, die in der Realität eine Abschwächung der Lidarstrahlen zur Folge hat.

Außerdem wird die Geschwindigkeit des Scanners nicht berücksichtigt. Bei dynamischen Szenarien ist die Punktwolke des realen Sensors breiter oder schmäler als das eigentliche Objekt, in der Simulation wird stattdessen das Szenario während des Scans angehalten. Bei einer maximalen Geschwindigkeit der Fahrzeuge von  $10 \frac{\text{km}}{\text{h}}$  in den Versuchen legt das Fahrzeug während eines Messscans ( $f = 12,5 \text{ Hz}$ ,  $\Delta t = 0,08 \text{ s}$ ) eine Strecke von bis zu 22 cm zurück. Die resultierende Abweichung zwischen realer und simulierter Punktwolke beeinflusst die wahrgenommenen Abmessungen der Objekte und damit insbesondere das Ergebnis der IoU-Metrik.

Dennoch kann in dieser Arbeit gezeigt werden, dass mit dem Objektlistenmodell ohne aufwändige Algorithmen Objektlisten generiert werden können, die lidartypische Merkmale aufweisen. Dabei können die Ergebnisse des Objektlistenmodells mit den konventionellen Algorithmen zur Erstellung der Objektliste verglichen werden, da beide auf demselben Sensormodell aufbauen. Eine generelle Beurteilung der Qualität des Objektlistenmodells ist schwierig, da es kaum Erfahrungswerte für Er-

---

gebnisse der Objektlisten-Metriken gibt, die eine Eignung zur Validierung autonomer Fahrfunktionen kennzeichnen.

---

## 8 Fazit und Ausblick

---

In dieser Arbeit wird ein neuartiges Objektlisten-Modell entwickelt, das ohne aufwendige Algorithmen zur Punktwolkenverarbeitung auskommt. Dabei werden neben den unverarbeiteten Punktwolken eines Sensormodells die Ground Truth Informationen der Simulationsumgebung verwendet, um lidartypische Objektlisten zu generieren. Das Modell ist als Modul in der Simulationsumgebung Vires VTD implementiert. Das Open Simulation Interface (OSI) wird dabei als generische Schnittstelle genutzt, sodass das Modell in unterschiedlichen Fahrsimulations-Frameworks zur Entwicklung automatisierter Fahrfunktionen genutzt werden kann.

Durch die Analyse realer Messdaten werden Merkmale lidartypischer Objektlisten sowie sensorspezifische Unsicherheiten des Ibeo Lux 2010 identifiziert. Mit diesen Informationen wird ein Objektlistenmodell erstellt, das die realen Objektlisten des Ibeo-Sensors simuliert. Dieses Modell wird anschließend mit Funktionen zur verbesserten Bestimmung der Objektposition, -geschwindigkeiten und -abmessungen erweitert, sodass auch modernere Sensoren simuliert werden können.

Zur Validierung des Objektlistenmodells werden reale Fahrversuche durchgeführt und anschließend nachsimuliert. Das Testscenario stellt eine alltägliche Situation an einer Kreuzung dar, bei der ein abbiegendes Fahrzeug vorübergehend durch ein anderes Fahrzeug verdeckt wird. Zum Vergleich der realen und simulierten Daten werden die Objektlisten anhand geeigneter Metriken bewertet. Es zeigt sich, dass die simulierten Objektlisten den realen bereits nahe kommen, aber dennoch Abweichungen bestehen. Die Hauptursache für die Unterschiede liegen in Vereinfachungen beim Sensormodell, das zur Generierung synthetischer Punktwolken genutzt wird. Besonders deutlich zeigen sich die Abweichungen während der Verdeckung des Zielobjektes. Unter dem Crafter verlaufende Lidarstrahlen in der Simulation führen dazu, dass das Objekt im Gegensatz zur Realität weiterhin gut wahrgenommen werden kann. Auch ohne Verdeckung existieren insbesondere bei der Objektposition und den geometrischen Abmessungen Abweichungen, die dadurch entstehen, dass die Reflexionseigenschaften von Objektoberflächen nicht berücksichtigt werden.

Die Ergebnisse der angewandten Metriken liegen im Bereich der Ergebnisse früher Arbeiten am Fachgebiet, die das gleiche Sensormodell zusammen mit konventionellen Algorithmen zur Erstellung der Objektliste nutzen. Allerdings ist ein direkter Vergleich aufgrund unterschiedlicher Fahrversuche nicht möglich.

Da für das erweiterte Modell kein realer Sensor zum Vergleich zur Verfügung steht, wird dessen Ausgabe mit dem Ibeo-Modell und der idealen Objektliste verglichen. Die Berechnung der Objektposition mithilfe der geschätzten Abmessungen liefert sehr gute Ergebnisse, die Objektposition liegt deutlich näher an der Ground Truth als beim Ibeo-Modell. Die Bestimmung der Relativgeschwindigkeit durch Ableitung eines ortsfesten Referenzpunktes am Objekt bringt noch keinen Vorteil gegenüber dem Ibeo-Modell. Mögliche Ursache ist der Sprung des Referenzpunktes, der kurzzeitig zu großen Abweichungen führt. Die Implementierung einer Erkennung dieser Sprünge könnten diese reduzieren. Zuletzt wird eine Bestimmung der Objektanmessungen aus aktuellen und vergangenen Messungen

---

implementiert. Dazu wird ein Qualitätswert berechnet, der angibt, wie gut die Abmessungen eines Objektes zum aktuellen Zeitpunkt bestimmt werden können. Im Track-Management werden dann die Abmessungen mit dem höchsten Qualitätswert gespeichert. Die Ergebnisse zeigen, dass die Funktion bereits funktioniert, der Qualitätswert aber noch Entwicklungspotential hat. Es hat sich gezeigt, dass der Qualitätswert zu stark vom Blickwinkel auf das Objekt abhängig ist, sodass das Ibeo-Modell in manchen Situationen die Länge oder Breite eines Objektes sehr genau bestimmt, das Tracking den Wert aber wegen des nicht optimalen Blickwinkels noch nicht übernimmt. Daher wird empfohlen, die Entfernung eines Objektes stärker im Qualitätswert zu gewichten. Alternativ ist auch eine Unterteilung des Sichtfeldes in verschiedene Bereiche denkbar, so dass beispielsweise die Entfernung zum Objekt bei geringen Distanzen eine höhere Gewichtung hat, während bei größeren Distanzen der Blickwinkel auf das Objekt den größeren Einfluss auf den Qualitätswert hat.

Es ist anzumerken, dass das implementierte Objektlistenmodell ein neuartiger Ansatz zur Generierung von lidartypischen Objektlisten ist, das von Grund auf entwickelt wurde. Daher konnten noch nicht alle Merkmale bzw. Unsicherheiten implementiert werden. Die wichtigste Aufgabe, um realistischere Ergebnisse zu erzielen, besteht in der Verbesserung des zugrundeliegenden Sensormodells. Zunächst sollten Bodenreflexionen aus der Punktwolke entfernt werden. Dazu muss nicht zwingend das Punktwolkenmodell bearbeitet werden. Es bietet sich an, die Bodenreflexionen in einem Modul herauszufiltern, das dem Objektlistenmodell vorgelagert ist. Zusätzlich könnte dort eine Filterung basierend auf der Form und Ausrichtung der Objekte erfolgen, sodass Reflexionen eliminiert werden, die in der Realität den Empfänger des Sensors nicht mehr erreichen. Eine weitere zu lösende Aufgabe ist die Klassifizierung von Objekten. Dazu empfiehlt sich eine merkmalsbasierte Klassifizierung, die anhand der wichtigsten Eigenschaften wie Länge, Breite und Geschwindigkeit den Objekten eine Klasse zuordnet. Eine weitere Unsicherheit, die in das Modell implementiert werden könnte, ist die Trennfähigkeit des Sensors.

Insgesamt zeigt sich, dass in der Entwicklung von Objektlistenmodellen großes Potential liegt, um die Rechenzeit von Simulationen zu verringern. Gleichzeitig sind Erfahrungswerte für die Ergebnisse von Metriken für Objektlisten nötig, um die Qualität des Modells bewerten zu können. Für einen Kompromiss aus niedriger Rechenzeit und möglichst realistischen Ergebnissen ist eine Kombination des Objektlistenmodells mit konventionellen Algorithmen denkbar, bei der beispielsweise das Objektlistenmodell die Segmentierung, Bestimmung der Position und Abmessungen durchführt und ein herkömmlicher Algorithmus für die Klassifizierung verwendet wird.

## A Anhang

### A.1 Zeitplan

Termin/Aufgaben	Monat	Dezember				Januar				Februar				März				April				Mai				Juni					
	KW	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Zwischenkolloquium Abgabe Abschlusskolloquium																															
Einarbeitung																															
Funktionsweise Lidarsensoren		X	X																												
Sensormodellierung		X	X	X																											
Vires VTD		X	X	X	X																										
Open Simulation Interface		X	X	X																											
Hauptphase																															
Merkmale lidartypischer Objektlisten identifizieren					X	X	X	X	X																						
Ansätze zur Generierung lidartypischer Objektlisten								X	X	X	X	X																			
Implementierung ideales Modell												X	X																		
Implementierung Ibeo-Modell												X	X	X																	
Implementierung erweitertes Modell																X	X	X	X												
Fahrversuche in Griesheim																			X												
Simulation der Fahrversuche																				X				X	X						
Evaluation																						X	X	X							
Ausarbeitung																															
Latex-Vorlage anpassen				X	X																										
Präsentation Zwischenkolloquium									X	X																					
Ausarbeitung schreiben				X	X														X						X	X					
Korrektur und Formatierung																									X	X					
Präsentation Abschlusskolloquium																										X	X				

Abbildung A-1: Soll-Zeitplan

Monat		Dezember				Januar				Februar				März				April				Mai				Juni						
KW		49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
Termin/Aufgaben																																
Zwischenkolloquium																																
Abgabe																																
Abschlusskolloquium																																
Einarbeitung																																
Funktionsweise Lidarsensoren		X	X																													
Sensormodellierung		X	X	X																												
Vires VTD		X	X	X	X	X	X																									
Open Simulation Interface		X	X	X																												
Hauptphase																																
Merkmale lidartypischer Objektlisten identifizieren				X	X	X	X	X																								
Ansätze zur Generierung lidartypischer Objektlisten							X	X	X	X																						
Implementierung ideales Modell												X	X																			
Implementierung Ibeo-Modell												X	X																			
Implementierung erweitertes Modell												X	X																			
Fahrversuche in Griesheim																X	X	X			X	X	X	X	X							
Simulation der Fahrversuche																						X	X	X	X	X						
Evaluation																						X	X	X	X	X						
Ausarbeitung																																
Latex-Vorlage anpassen				X	X																											
Präsentation Zwischenkolloquium										X																						
Ausarbeitung schreiben				X	X	X	X	X											X	X	X	X	X	X	X	X	X					
Korrektur und Formatierung																																
Präsentation Abschlusskolloquium																												X	X			X

Abbildung A-2: Ist-Zeitplan

## A.2 Ergebnisse des Ibeo Modells

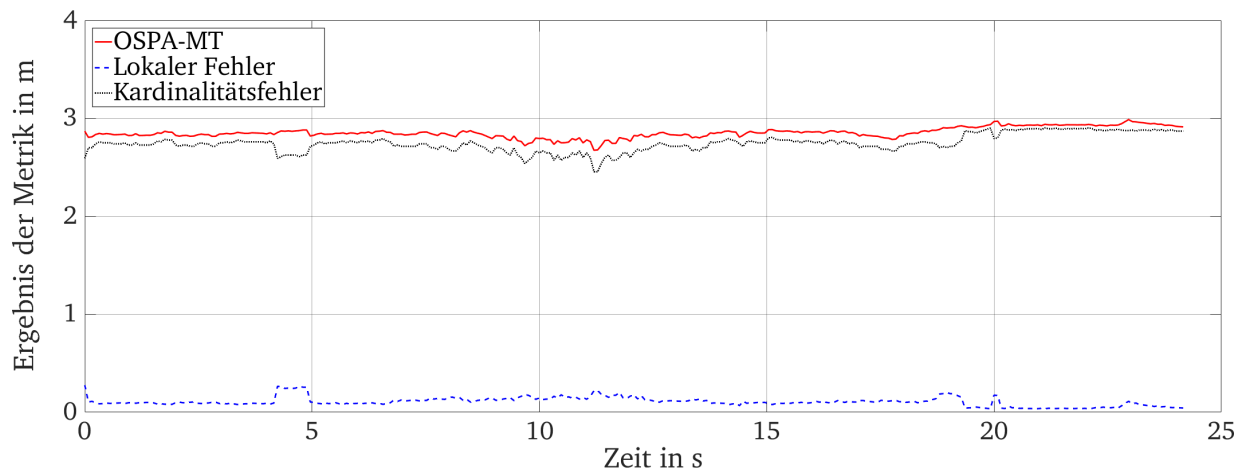


Abbildung A-3: OSPA-MT Metrik für reale Daten und Ibeo-Modell, Versuch 2

## A.3 Ergebnisse des erweiterten Modells, Versuch 1

### A.3.1 Erweitertes Modell mit verbesserter Positionsbestimmung

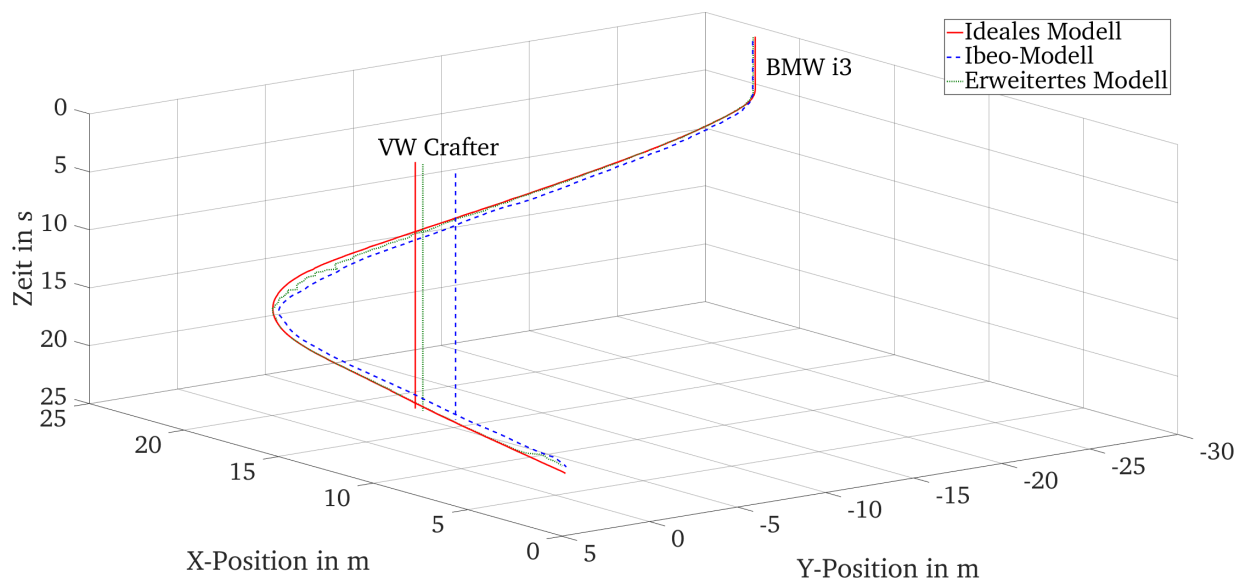


Abbildung A-4: Trajektorien von idealem Modell, Ibeo-Modell und verbessertem Modell, Versuch 1

Tabelle A-1: RMSE für erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
Geom. Mittelpunkt in m	0,875	0,245	2,137	0,403



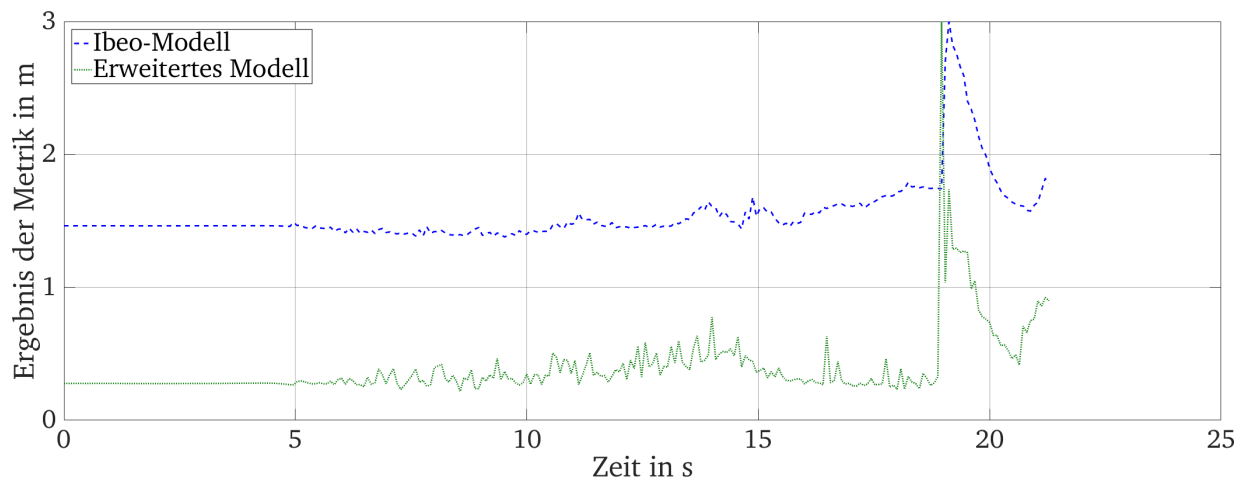


Abbildung A-5: OSPA-MT-Metrik für Ibeo-Modell und erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1

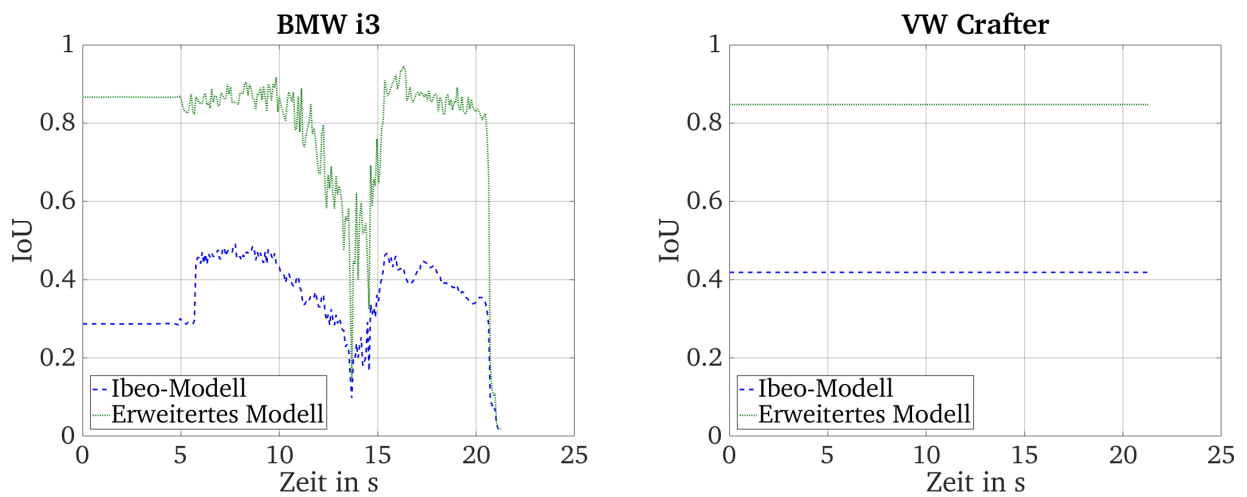


Abbildung A-6: IoU für Ibeo-Modell und erweitertes Modell mit verbesserter Positionsbestimmung, Versuch 1

### A.3.2 Erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung

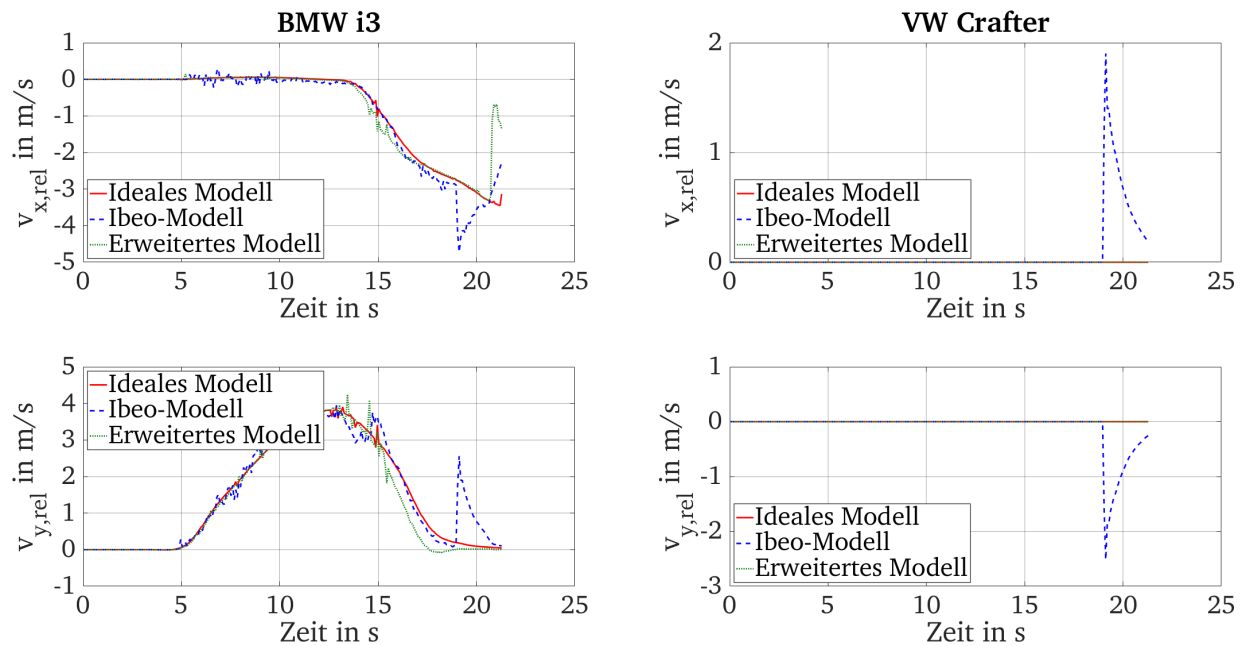


Abbildung A-7: Relativgeschwindigkeiten von idealem Modell, Ibeo-Modell und verbessertem Modell, Versuch 1

Tabelle A-2: RMSE für erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 1

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
$v_x$ in $\frac{m}{s}$	0,304	0,406	0,272	0,000
$v_y$ in $\frac{m}{s}$	0,360	0,258	0,359	0,000

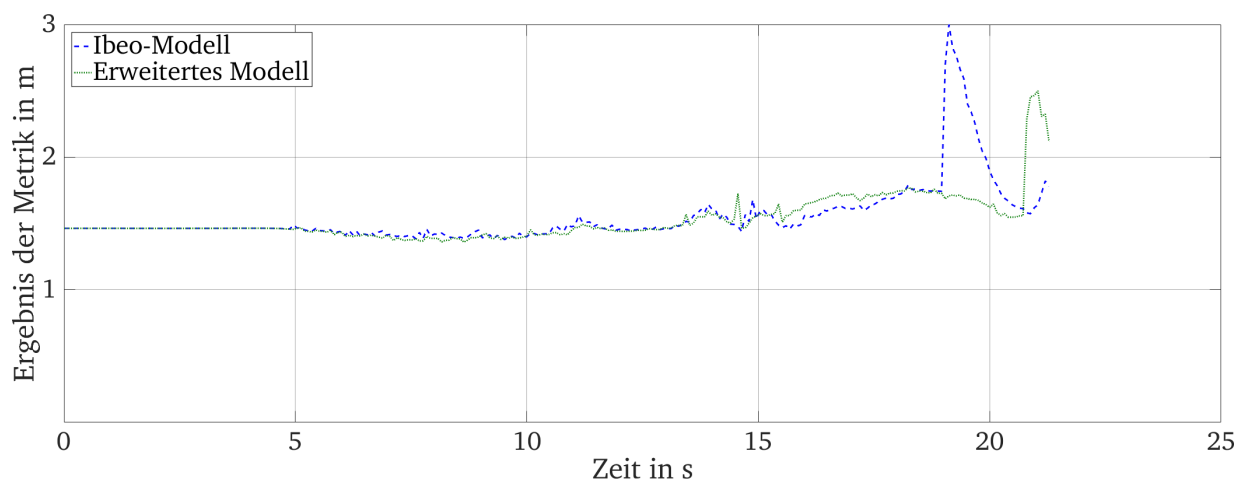


Abbildung A-8: OSPA-MT-Metrik für Ibeo-Modell und erweitertes Modell mit verbesserter Geschwindigkeitsbestimmung, Versuch 1

## A.4 Erweitertes Modell mit verbesserter Bestimmung der Objektmessungen

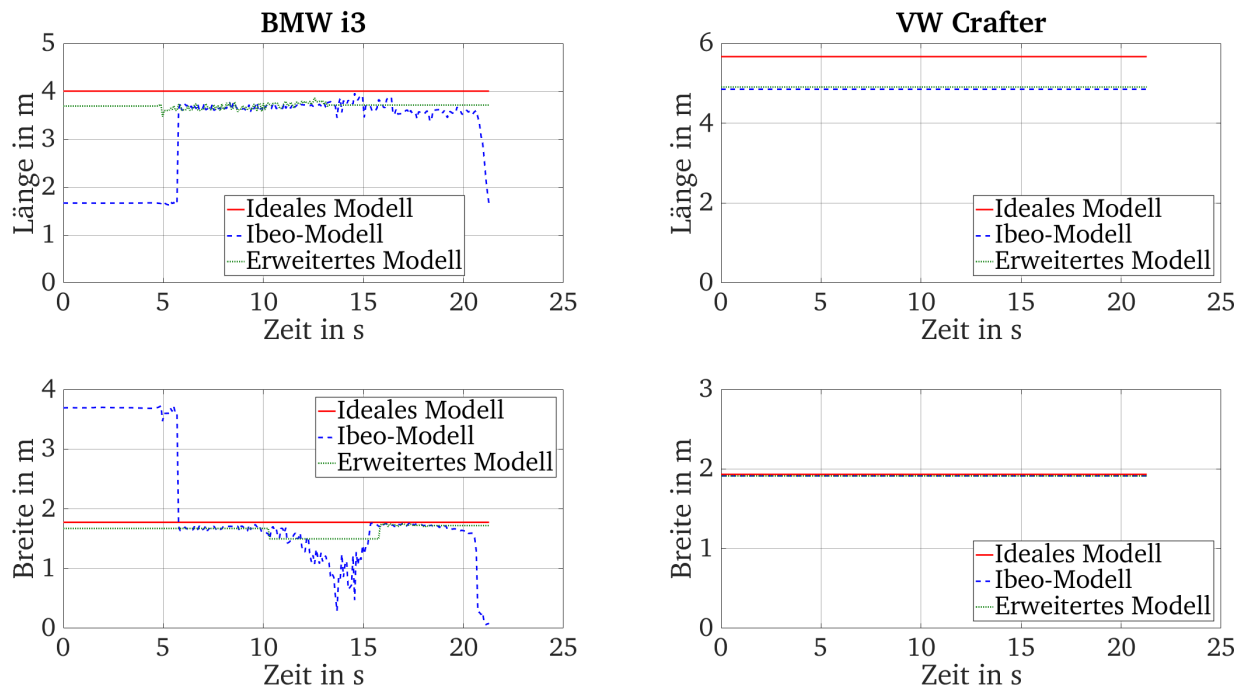


Abbildung A-9: Objektmessungen von Ibeo-Modell und verbessertem Modell, Versuch 1

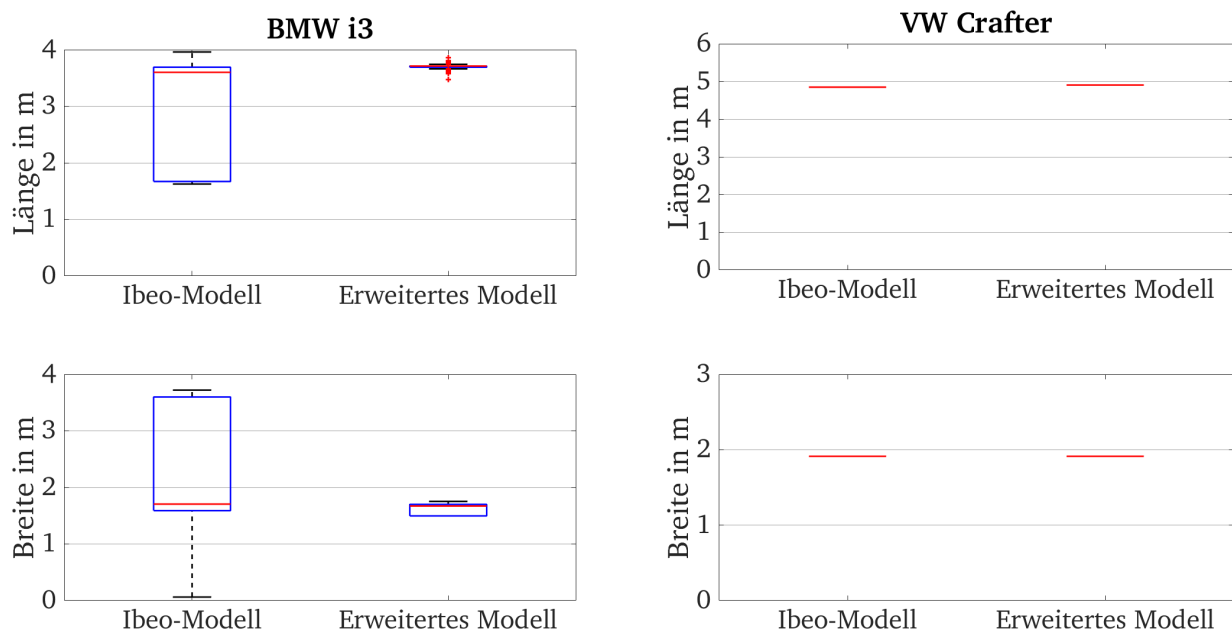


Abbildung A-10: Objektmessungen von Ibeo-Modell und verbessertem Modell als Boxplots, Versuch 1

Tabelle A-3: RMSE für erweitertes Modell mit verbesserter Bestimmung der Objektanmessungen, Versuch 1

RMSE	BMW i3		VW Crafter	
	Ibeo Modell	Erweitertes Modell	Ibeo Modell	Erweitertes Modell
Länge in m	0,468	0,302	0,817	0,761
Breite in m	0,488	0,177	0,019	0,019

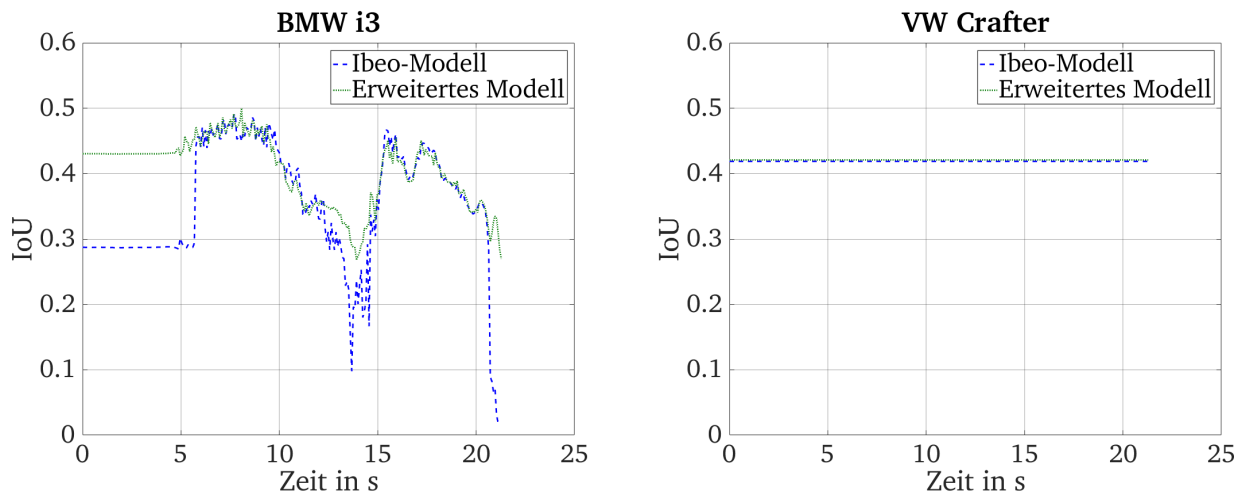


Abbildung A-11: IoU für Ibeo-Modell und erweitertes Modell mit verbesserter Bestimmung der Objektanmessungen, Versuch 2

---

## Literaturverzeichnis

---

**Bert, F.: Identifikation von Merkmalen für die Objekterkennung mit lernenden LiDAR-Algorithmen (2017)**

Bert, Felix: „Identifikation von Merkmalen für die Objekterkennung mit lernenden LiDAR-Algorithmen“, Studienarbeit, Technische Universität Darmstadt, (2017)

**Dietmeier, K.: Predicting of Machine Perception for Automated Driving (2016)**

Dietmeier, Klaus: Predicting of Machine Perception for Automated Driving. In: Maurer M., Gerdes J., Lenz B., Winner H. (eds) Autonomous Driving, Springer, (2016)

**ENABLE-S3: Homepage Forschungsprojekt ENABLE-S3 (2018)**

ENABLE-S3: Homepage Forschungsprojekt ENABLE-S3, URL: <https://www.enable-s3.eu/>, (besucht am 06.01.2019), (2018)

**Friedmann, L.: Open Simulation Interface Wiki (2017)**

Friedmann, Ludwig: Open Simulation Interface Wiki, URL: <https://github.com/OpenSimulationInterface/open-simulation-interface/wiki>, (besucht am 04.01.2019), (2017)

**Fürstenberg, K. C.: Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner (2009)**

Fürstenberg, Kay Christian: „Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner“, Dissertation, Universität Ulm, (2009)

**Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015)**

Gotzig, Heinrich; Geduld, Georg: LIDAR-Sensorik. In: Winner H., Hakuli S., Lotz F., Singer C. Handbuch Fahrerassistenzsysteme, Springer Vieweg, (2015)

**Granström, K. et al.: Extended Object Tracking: Introduction, Overview and Applications (2017)**

Granström, Karl; Baum, Marcus; Reuter, Stephan: „Extended Object Tracking: Introduction, Overview and Applications“, (2017)

**Hanke, T. et al.: Open Simulation Interface (2017)**

Hanke, Timo; Hirsenkorn, Nils; van Driesten, Carlo; Garcia Ramos, Pilar; Schiementz, Mark; Schneider, Sebastian: Open Simulation Interface: A generic interface for the environment perception of automated driving functions in virtual scenarios. URL: <http://www.hot.ei.tum.de/forschung/automotive-veroeffentlichungen/>, (besucht am 04.01.2019), (2017)

**Huch, S.: Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation (2018)**

Huch, Sebastian: „Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation“, Masterthesis, Technische Universität Darmstadt, (2018)

**Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® Laserscanner (2015)**

Ibeo Automotive Systems GmbH: „Operating Manual ibeo LUX 2010® Laserscanner“, (2015)

**Jin, G.: Enhancement of an ideal Lidar sensor model with identified physical effects (2018)**

Jin, Geng: „Enhancement of an ideal Lidar sensor model with identified physical effects“, Mastertesis, Technische Universität Darmstadt, (2018)

**Klasing, K. et al.: A Clustering Method for Efficient Segmentation of 3D Laser Data (2008)**

Klasing, Klaas; Wollherr, Dirk; Buss, Martin: „A Clustering Method for Efficient Segmentation of 3D Laser Data“, in: *IEEE International Conference on Robotics and Automation*, (2008)

---

**Knerr, J.: Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018)**

Knerr, Jonathan: „Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle“, Masterthesis, Technische Universität Darmstadt, (2018)

**Morton, P. et al.: An evaluation of dynamic object tracking with 3D LIDAR (2011)**

Morton, P.; Douillard, B.; Underwood, J.: „An evaluation of dynamic object tracking with 3D LIDAR“, (2011)

**Ngo, A.: Erweiterung eines LiDAR Sensormodells für simulationsbasiertes Entwickeln und Testen von Objekterkennungsalgorithmen (2018)**

Ngo, Anthony: „Erweiterung eines LiDAR Sensormodells für simulationsbasiertes Entwickeln und Testen von Objekterkennungsalgorithmen“, Masterthesis, Technische Universität Darmstadt, (2018)

**Nguyen, A. et al.: 3D Point Cloud Segmentation: A survey (2013)**

Nguyen, Anh; Le, Bac: „3D Point Cloud Segmentation: A survey“, in: *IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, (2013)

**Pegasus: Homepage Forschungsprojekt PEGASUS (2018)**

Pegasus: Homepage Forschungsprojekt PEGASUS, URL: <https://www.pegasusprojekt.de/de/>, (besucht am 06.01.2019), (2018)

**SAE International: J3016 (2018)**

SAE International: „J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles“, (2018)

**Schuler, K. et al.: Extraction of Virtual Scattering Centers of Vehicles by Ray-Tracing Simulations (2008)**

Schuler, Karin; Becker, Denis; Wiesbeck, Werner: „Extraction of Virtual Scattering Centers of Vehicles by Ray-Tracing Simulations“, in: *IEEE Transactions on Antennas and Propagation*, (2008)

**Sovani, S.: Simulation Accelerates Development of Autonomous Driving (2017)**

Sovani, Sandeep: „Simulation Accelerates Development of Autonomous Driving“, (2017)

**Stäcker, L.: Entwicklung eines Informationssystems zum Segmentieren und Tracking von Objekten auf Basis von LiDAR-Sensordaten für Autonomes Fahren (2017)**

Stäcker, Lukas: „Entwicklung eines Informationssystems zum Segmentieren und Tracking von Objekten auf Basis von LiDAR-Sensordaten für Autonomes Fahren“, Masterthesis, Technische Universität Darmstadt, (2017)

**Vijayendra, V. B.: Refinement of a Virtual Environment Representation to conduct validation tests for Automated driving (2018)**

Vijayendra, Vishwas Bangalore: „Refinement of a Virtual Environment Representation to conduct validation tests for Automated driving“, Masterthesis, Technische Universität Darmstadt, (2018)

**Vu, T. et al.: Optimal Subpattern Assignment Metric for Multiple Tracks (2018)**

Vu, Tuyet; Evans, Rob: „Optimal Subpattern Assignment Metric for Multiple Tracks“, (2018)

**Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2015)**

Wachenfeld, Walther; Winner, Hermann: The Release of Autonomous Vehicles. In: Maurer M., Gerdes C., Lenz B., Winner H. *Autonomous Driving*, Springer Vieweg, (2015)

**Waymo: Million of miles driven (2018)**

Waymo: Million of miles driven, URL: <https://waymo.com/ontheroad/>, (besucht am 06.01.2019), (2018)

---

**Winner, H.: How to Address the Approval Trap for Autonomous Vehicles (2015)**

Winner, Hermann: „How to Address the Approval Trap for Autonomous Vehicles“, (2015)

**Winner, H.: Quo vadis, FAS? (2015)**

Winner, Hermann: Quo vadis, FAS? In: Winner H., Hakuli S., Lotz F., Singer C. Handbuch Fahrerassistenzsysteme, Springer Vieweg, (2015)

**Yungeun, C. et al.: Fast Point Cloud Segmentation for an Intelligent Vehicle Using sweeping 2D laser scanners (2012)**

Yungeun, Choe; Seunguk, Ahn; Myung, Jin Chung: „Fast Point Cloud Segmentation for an Intelligent Vehicle Using sweeping 2D laser scanners“, in: *IEEE International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, (2012)

**Zeit Online: Tesla-Unfall: Ermittler machen Autopilot für tödlichen Unfall mitverantwortlich (2017)**

Zeit Online: Tesla-Unfall: Ermittler machen Autopilot für tödlichen Unfall mitverantwortlich, URL: <https://www.zeit.de/wirtschaft/unternehmen/2017-09/tesla-unfall-autopilot-mitverantwortlich>, (besucht am 06.01.2019), (2017)

**Zhang, Y.: Implementation of an enhanced Lidar Tracking algorithm for automated driving (2018)**

Zhang, Yanni: „Implementation of an enhanced Lidar Tracking algorithm for automated driving“, Masterthesis, Technische Universität Darmstadt, (2018)

**Zhou, Y. et al.: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection (2017)**

Zhou, Yin; Tuzel, Oncel: „VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection“, (2017)